

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«Київський політехнічний інститут імені Ігоря Сікорського»
Інженерно-хімічний факультет
Кафедра “Автоматизації хімічних виробництв”

МАГІСТЕРСЬКА ДИСЕРТАЦІЯ
на тему: «Система керування процесом графітування у печах
Ачесона»

Виконав:
Корж А. П.
студент VI курсу, IXФ
група ЛА-61м
залікова книжка № ЛА-6107м

Керівник магістерської дисертації:

к. т. н., доцент Жученко О. А.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Інженерно-хімічний факультет
(повна назва інституту/факультету)

Автоматизації хімічних виробництв
(повна назва кафедри)

«На правах рукопису»
УДК 661.666.2

«До захисту допущено»

Завідувач кафедри

_____ Жученко А. І.
(підпис) (ініціали, прізвище)

“ _____ ” _____ 2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 151 – Автоматизація та комп'ютерно-інтегровані технології
(код і назва)

на тему: «Система керування процесом графітування у печах Ачесона»

Виконав: студент VI курсу, групи ЛА-61м
(шифр групи)

_____ Корж Андрій Петрович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к. т. н., доцент Жученко О. А. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент д.т.н., професор Панов Є. М. _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ – 2018 року

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Інститут (факультет) Інженерно-хімічний факультет
(повна назва)

Кафедра Автоматизації хімічних виробництв
(повна назва)

Рівень вищої освіти – другий (магістерський)

Спеціальність 151- Автоматизація та комп'ютерно-інтегровані технології
(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри

Жученко А. І.
(підпис) (ініціали, прізвище)

«26» березня 2018 р.

**ЗАВДАННЯ
на магістерську дисертацію студенту**

Коржа Андрія Петровича
(прізвище, ім'я, по батькові)

1. Тема дисертації «Система керування процесом графітування у печах Ачесона»,

науковий керівник дисертації доц., к. т. н. Жученко О. А
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «21» березня 2018 р. №979-с

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження Технологічний процес графітування у печах Ачесона

4. Предмет дослідження Система керування процесом графітування у печах Ачесона

5. Перелік завдань, які потрібно розробити Навести опис та схему технологічного процесу, розробити математичну модель процесу графітування у печах Ачесона, дослідити температурні поля, створити спрощену модель процесу, розробити та дослідити систему керування процесом на основі ступеня графітування та оптимального керування з передчасним вимкненням живлення

6. Орієнтовний перелік ілюстративного матеріалу _____

7. Орієнтовний перелік публікацій «Класифікація електродної продукції та її застосування», «Показники якості графітованої продукції», «Аналіз систем керування процесом графітування у печах ачесона», «Постановка задачі вибору системи керування процесом графітації у печах Ачесона», «Спрощення математичної моделі процесу графітування у печах ачесона».

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання 29 березня 2018р

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Опис виробництва	12.02.2018	
2	Розробка математичної моделі процесу	20.02.2018	
3	Аналіз існуючих методів спрощення моделі	05.03.2018	
4	Створення спрощеної моделі процесу	07.04.2018	
5	Розробка та дослідження системи керування процесом	21.04.2018	
6	Розробка стартап-проекту	01.05.2018	
7	Оформлення матеріалів до магістерської дисертації	12.05.2018	

Студент

_____ (підпис)

Корж А. П.

(ініціали, прізвище)

Науковий керівник дисертації

_____ (підпис)

Жученко О. А.

(ініціали, прізвище)

РЕФЕРАТ

Дана магістерська дисертація містить теоретичні відомості із розробки математичної моделі печі графітування, її спрощення, аналіз та розробку системи керування піччю Ачесона. Дисертація оформлена у пояснювальну записку об'ємом 134 сторінок. Пояснювальна записка містить рисунки, таблиці, та 45 літературних джерел в тому числі одну публікацію автора до дисертації.

В даній роботі об'єктом дослідження є процес графітування. Предметом дослідження є математична модель печі графітування, способи її спрощення.

У магістерській дисертації досліджено особливості роботи печі Ачесона як об'єкта керування, для якої розроблено системи керування на базі ступення графітування та оптимального керування з передчасним вимкненням живлення, а також мовою *C#* та *WPF* розроблено прикладне програмне забезпечення для імітації роботи печі Ачесона та можливістю керувати процесом.

У магістерській дисертації на базі програмного засобу *MATLAB* розроблено скрипт для спрощення складних моделей за експериментальними даними, розроблено систему керування процесом на базі ступення графітування та оптимального керування з передчасним вимкненням живлення.

Розроблено стартап-проект присвячено системі керування процесом графітування у печах Ачесона

Ключові терміни: ГРАФІТУВАННЯ, ПІЧ АЧЕСОНА, *ANSYS*, *MATLAB*, СПРОЩЕННЯ МОДЕЛІ, СТУПІНЬ ГРАФІТУВАННЯ, *C#*, *.Net*, *WPF*.

ABSTRACT

This master's dissertation contains theoretical information on the development of mathematical model of the graphite furnace, its simplification, analysis and development of the control system of the Acheson furnace. The thesis is presented in an explanatory note of 134 pages. The explanatory note contains drawings, tables, and 45 literary sources, including one publication of the author to the dissertation.

In this paper, the object of the study is the graphing process. The subject of the study is a mathematical model of the furnace graphing, ways of its simplification.

In the master's thesis the features of the Acheson furnace operation as a control object were developed for which a control system based on the degree of graphing and optimal control with premature power failure was developed, as well as the C # language and WPF application software was developed for simulating the work of Acheson's furnace and the ability to control the process .

In the master's thesis on the basis of the software MATLAB developed a script to simplify complex models of experimental data, developed a process control system based on the degree of graphing and optimal control with premature power failure.

A startup project is developed on the process of controlling the carbonization of carbon products.

Key terms: GRAPHITATION, FURNACE ACHESON, ANSYS, MATLAB, CLEAR MODEL, GRAPHING STEP, C #, .Net, WPF.

ЗМІСТ

ВСТУП.....	9
1. АНАЛІЗ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ГРАФІТУВАННЯ ВУГЛЕЦЕВИХ ВИРОБІВ.....	11
1.1. Опис технологічної схеми електродного виробництва графітованої електродної продукції.....	11
1.2. Конструкція печі графітування	18
1.3. Піч графітування як об'єкт автоматизації.....	20
2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ У ПЕЧІ АЧЕСОНА.....	26
2.1. Існуючі математичні моделі процесів у печі Ачесона	26
2.2. Способи виміру температури в печі графітування.....	31
2.3. Математичне моделювання фізичних полів під час графітування заготовок у печі Ачесона	34
2.4. Вибір методу чисельного розрахунку.....	45
2.5. Дослідження температурного печі Ачесона.....	46
3. СПРОЩЕННЯ МОДЕЛІ ПРОЦЕСУ	49
3.1. Спрощення моделі процесу	49
3.2. Аналіз існуючих методів спрощення моделі	49
3.3. Розроблення спрощеної моделі за допомогою методу Фур'є	50
3.3.1. Метод Фур'є.....	50
3.3.2. Визначення базисних векторів та коефіцієнтів Фур'є.....	51
3.3.3 Ідентифікація залежностей між отриманими значеннями коефіцієнтів Фур'є та вхідними сигналами.....	56
3.3.4 Дослідження якості роботи спрощених моделей	58
4. СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ПІЧЧЮ ГРАФІТУВАННЯ.....	62
4.1. Керування процесом графітуванням у печах Ачесона	62
4.2. Способи керування графітуванням у печах Ачесона	62
4.3. Проектування системи керування піччю графітування	63
5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КЕРУВАННЯ ПРОЦЕСОМ ГРАФІТУВАННЯ У ПЕЧАХ АЧЕСОНА	72
5.1. Актуальність розробки програмного забезпечення.....	72
5.2. Проектування програмного забезпечення для процесу графітування у печах Ачесона ..	73
5.3. Вибір середовища розробки програмного забезпечення	74
5.4. Розробка програмного забезпечення.....	76
5.5. Розробка програмного забезпечення для віддаленого доступу.....	81

6. СТВОРЕННЯ СТАРТАП ПРОЕКТУ	83
6.1 Опис ідеї стартап-проекту.....	83
6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту.....	84
6.3 Технологічна здійсненність ідеї проекту.....	84
6.4 Характеристика потенційних клієнтів	85
6.5 Фактори загроз	86
6.6. Фактори можливостей	87
6.7. Ступеневий аналіз конкуренції на ринку	87
6.8. Аналіз конкуренції в галузі за М. Портером.....	88
6.9. Обґрунтування факторів конкурентоспроможності.....	89
6.10. Порівняльний аналіз сильних та слабких сторін «Процесу графітації за допомогою печі Ачесона»	90
6.11. SWOT – аналіз стартап-проекту	90
6.12. Вибір цільових груп потенційних споживачів.....	91
6.13. Визначення базової стратегії розвитку	92
6.14. Визначення базової стратегії конкурентної поведінки	92
6.15. Визначення стратегії позиціонування.....	93
6.16. Визначення ключових переваг концепції потенційного товару	94
6.17. Визначення меж встановлення ціни.....	94
6.18. Формування системи збуту.....	94
6.19. Концепція маркетингових комунікацій	95
6.20. Опис трьох рівнів моделі товару	96
ВИСНОВОК	97
ЛІТЕРАТУРА	98
ДОДАТКИ	104
Додаток 1 – Лістинг програмного забезпечення програми мовою <i>C#.Net</i>	104
Додаток 2 – Лістинг програмного забезпечення програми мовою <i>WPF .Net</i>	119

ВСТУП

Металургійна промисловість почала збільшувати обсяги виробництва та перейшла на використання нового високотехнологічного обладнання. Поширеними стали нові види електродної продукції - великогабаритні електроди, графітові подові блоки і аноди і т.д., адже мають підвищені характеристики експлуатації.

При виконанні електродних виробів застосовується високоякісний кокс нафтовий голчастої структури, а питомі витрати електроенергії при їхньому графітуванні в печах непрямого нагріву (печі Ачесона) зростають. Виробництво графітованої продукції складне, багатостадійне та дуже енергоємне. При цьому на заключну стадію - графітацію - припадає до 80 % загальних енерговитрат. Особлива увага при виробництві електродних виробів з підвищеними експлуатаційними характеристиками надається забезпеченню однорідності їхніх фізичних властивостей, адже широкий асортимент продукції, різний рівень параметрів якості, закладених в технічних вимогах до готової продукції, а також різні технічні характеристики печей графітування Ачесона зумовлюють потребу розробки індивідуальних технологічних регламентів процесу графітування. Особливе місце серед електродної продукції посідає графітована електродна продукція, оскільки має високі показники якості, наприклад, висока міцність, теплопровідність, низький питомий опір тощо.

Процес виготовлення графітованої електродної продукції не складний, але дуже енергоємний, а тому – надзвичайно затратний. Одним із найважливіших апаратів у всій технологічній лінії виготовлення графітованої електродної продукції є піч графітування, або піч Ачесона.

Піч Ачесона – це електропіч, із досить великим споживанням електроенергії. Тому, навіть не значне зменшення питомої витрати електроенергії призвело б до суттєвої економії та здешевлення готової продукції.

З іншого боку, піч Ачесона, як об'єкт автоматизації, є надзвичайно складним, через свою специфіку процесів, що відбуваються в апараті під час кампанії графітування. Це значно ускладнює розробку відповідної системи керування.

1. АНАЛІЗ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ГРАФІТУВАННЯ ВУГЛЕЦЕВИХ ВИРОБІВ

1.1. Опис технологічної схеми електродного виробництва графітованої електродної продукції

Принципова технологічна схема електродного виробництва складається із ряду основних операцій: прийом та підготовка сировинних матеріалів, приготування електродної маси, формування заготовок, обпалення, просякнення, графітування та механічна обробка заготовок.

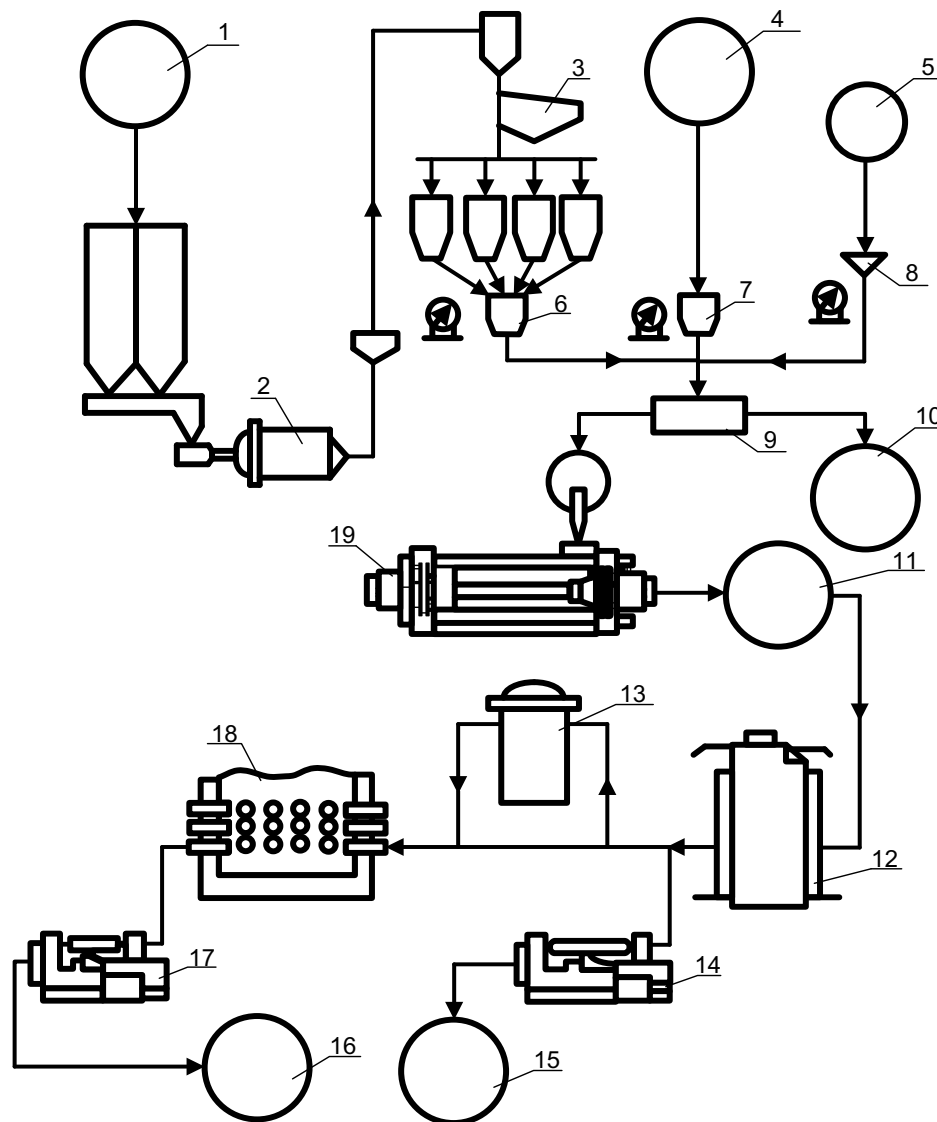


Рис.1.1 – Технологічна схема електродного виробництва: 1 – Склад сировини, 2 – дроблення та позмелення, 3 – розсів, 4 – склад пеку, 5 – склад добавок, 6 – дозування шихти, 7 – дозування пеку, 8 – дозування добавок, 9 – змішування, 10 – товарні маси, 11 – склад «зелених» заготовок, 12- обпалення, 13 – просякнення, 14 – механічна обробка вугільної продукції, 15- склад вугільної продукції, 16 – склад графітованої продукції, 17 – механічна обробка графітованої продукції, 18 – графітування, 19 – пресування.

За цією схемою виробляють графітовану продукцію [1]. При виготовленні вугільних виробів виключається етап графітування, а товарні маси після змішування і брикетування відправляються споживач.

Як показує величезний досвід заводів, всі технологічні операції мають велике значення для забезпечення високого виходу якісної електродної продукції.

Дуже важливо вірно організувати роздільне надходження і закрите зберігання сировинних матеріалів роздільних за своїми властивостями. Це дозволяє забезпечити точне дозування сировини на наступних стадіях виробництва, що є основою для отримання виробів стабільної якості.

Дотримання режимів змішування, охолодження та пресування електродної маси і використання мундштуків заданої геометрії за формування дає можливість отримувати заготовки з високим виходом придатності. Під час обпалення, найдовшому із процесів, заготовки зазнають значні термохімічні зміни, в результаті яких формуються основні властивості майбутніх виробів. Для забезпечення збереження форми та цілісності заготовок, отримання міцного та однорідного по структурі обпаленого матеріалу, необхідно строго дотримуватися всіх технологічних параметрів обпалення.

При наступній термічній обробці заготовок у печах графітації, вони набувають високу електро- та теплопровідність. У даному випадку необхідне дотримання не лише електричних режимів графітування, але і схем завантаження ті упаковки печі пересипними та теплоізоляційними матеріалами.

Точна обробка ніпеля та ніпельного гнізда досягається в процесі механічної обробки і в значній мірі впливає на питому витрату електродів у споживачів.

В отриманні якісних виробів електродного виробництва важливу роль грає контроль, дотримання технологічного режиму і властивостей сировини, напівпродуктів та готової продукції на кожному етапі [2].

Сировиною для отримання штучного графіту слугують продукти нафтової та кам'яновугільної промисловості: піролізні та крекінгові нафтові кокси, антрацити та термоантрацити, пекові кокси. Ці продукти розрізняються за способом їх отримання, по вмісту вуглецю, домішок, а також фізико – хімічним та механічним властивостям.

Нафтові кокси – це продукт коксування нафтових залишків, наприклад, гудронів та крекінг-залишків. Їх властивості залежать не тільки від складу вихідної нафти, але і від способів її переробки. Нафтові кокси – високоякісний вуглецевий матеріал, що містить до 98% вуглецю та мало домішок. Вони мають темний колір, блищать та мають великі пори.

Пековий чи смоляний кокс отримують при переробці кам'яновугільної смоли. За своїми перевагами для виробництва штучного графіту він близький до нафтового коксу. Пековий кокс містить меншу кількість летких речовин. Практика показала, що графітовані електроди, які виготовлені на основі пекового коксу, термостійкі та володіють великою механічною міцністю й достатньою електропровідністю.

Кокси, які використовуються у виробництві вуглеграфітових матеріалів, є неоднорідними за своєю структурою. У них виявлені дві яскраво виражені складові – точкова та волоконна. Виділення цих структурних складових з шматків коксу та вивчення їх властивостей показало, що вони володіють різними механічними та фізико-хімічними властивостями. Ці дані вказують на необхідність диференційованого підходу до вибору сировини певної структури для того чи іншого виду продукції.

Характеристика сировини варто вважати неповною якщо не згадувати про в'язучі матеріали, позаяк без них в наш час неможливо уявити виробництво монолітних вуглеграфітових виробів. У якості в'язучого матеріалу слугують кам'яновугільні пеки з різними температурами розм'якшення та смоло-пеки. Пеки отримують з кам'яновугільної смоли після виділення з неї антраценових масел в перегінних кубах. Від ступеню виділення масел залежить склад пеку і температура розм'якшення цієї в'язкої та чорної маси. По температурі

розм'якшення, пеки поділяють на тверді, середні та м'які. У виробництві вуглеграфітових виробів найширше використовують середньотемпературний пек. Однак, зараз робиться все більше спроб використати у виробництві тугоплавкий пек.

Перед використанням, сировина проходить ряд підготовчих операцій. Спочатку іде операція попереднього дроблення. Сирі вуглецеві матеріали із складів постачаються до дробарок. Розмір шматків після дроблення сягає 40 – 45мм. Роздроблений вуглецевий матеріал надходить в печі прокалення, де з нього видаляються леткі речовини та волога. Густина та електропровідність матеріалу зростають. Зазвичай температура пропалювання підтримується не нижче температури обпалювання, але варто врахувати, що при температурах 773 – 1973 та 2673K прокалювання менш ефективне, ніж прокалювання за температури 1573K.

Прокалений матеріал після охолодження надходить у відділення розмелу, де він розмелюється до зерен розміром 0.5 – 10мм. В ситовому відділі за допомогою вібраційних та обертових сит він розсіюється по фракціям та зсипається у сортові бункери. Варто відзначити, що завдяки структурній неоднорідності коксів при розмеленні та просіюванні відбувається деякий натуральний поділ на волокнисту та точкову складову структури. В результаті цього окремі фракції виявляються збагаченими волокнистою складовою, що повинно враховуватись при складанні сумішей.

Якщо суміш вірно підібрана – це призводить в подальшому до збільшення механічної міцності виробів. Великі зерна у виробках грають роль скелету, а малі – наповнювача. Кожний вид вуглеграфітової продукції має свій рецепт виготовлення, що розроблений завдяки численним експериментам.

Відображення до вище написаного - слугує таблиця 1.1.

Із сортових дозувальних бункерів певна кількість кожної фракції та в'язуче (кам'яновугільний пек) подаються у змішувач. Гарячі змішувачі оснащені спеціальними паровими кожухами для підігріву. Шихта перемішується протягом 1 – 5 год. Температура маси сягає 425K. Готова електродна маса зі

змішувачів надходить в міксери, де вона охолоджується за допомогою вентиляторів до 373К. Тільки після цієї операції електродна маса надходить до пресового відділу.

Мікроскопічні дослідження показали, що за гарного змішування електродна маса характеризується рівномірним розподілом великих і малих компонентів наповнювача та однаковим розподілом пекових прошарків у масі. Якість маси залежить не тільки від вмісту в'язучого, але й від температури та часу змішування. Варто зазначити, що нерівномірне розподілення в'язучого і компонентів наповнювача призводить до дефектів макро- та мікробудови.

В пресовому відділі вуглецевій масі надають необхідну форму і отримують так звані «зелені» заготовки.

Пресовані маси виготовляють одним із двох способів:

- 1) формування в закриту матрицю;
- 2) продавлюванням через мундштук відповідної форми.

Важливо зазначити, що в обох випадках пресування в «зелених» заготовках виникає текстура. Вона не зникає навіть після високотемпературної обробки і проявляється в тому, що вуглеграфітові вироби, в залежності від напрямку по відношенню до осі пресування, мають різні механічні та теплофізичні властивості. Причини виникнення текстури у анізотричності зерен наповнювача і способі пресування.

У процесі пресування видовженні частинки коксу розташовуються своїми довгими осями перпендикулярно до напрямку прикладення сили. Це означає, що при формуванні вони розташовуються перпендикулярно до осі пресування, а при продавлюванні через мундштук – паралельно вісі пресування.

Після завантаження вуглецевої маси в циліндр пресу її ущільнюють. Для підтримання необхідної температури маси електродні преси обігрівають парою. Потужність пресів коливається в широких межах. Очевидно, що для пресування великогабаритних «зелених» заготовок, наприклад, вагою до 30 тон, необхідні преси.

Ущільнена електродна маса випресовується через мундштук за тиску $(175-350) \cdot 10^3 \text{ Н/м}^2$. У закриту прес-форму пресують при підвищених тисках. Вони складають близько $200 \cdot 10^4 \text{ Н/м}^2$ та вище. Вироби після пресування мають температуру близько 350 K і можуть деформуватись під дією власної ваги. Для попередження деформації їх охолоджують до затвердіння у водяній ванні.

Застиглі «зелені» заготовки завантажують в кільцеву багатокамерну газову піч та обпалюють без доступу повітря, тому, вироби, що розміщені в касетах печі, засипають, а зверху піч замурують. У якості засипки часто застосовують гранульовану коксо-піщану суміш. Під склепінням печі в полум'яних колодязях спалюється генераторний газ. Продукти горіння через спеціальні димоходи надходять із однієї камери в іншу і розігрівають її. Вироби, що обпалюються, не контактують з продуктами горіння, які, віддавши тепло, викидаються в атмосферу через димову трубу. Температура засипки, що вимірюється за допомогою термопар сягає 1200 – 1500 K. Операція обпалення дуже довга і триває до 20 – 40 діб.

Вивантаження обпалених виробів здійснюється лише після повного охолодження печі. Після вивантаження придатні вироби піддають механічному очищенню від спікшихся і прококованих часток засипки. Для надання їм необхідних розмірів і форми їх обробляють на токарних станках.

Отримані в процесі обпалення вуглецеві вироби називають вугільними. Частина з них надходить до заводів-споживачів, а інша частина зазнає високотемпературної обробки в електричних печах графітування. У окремих випадках вугільні вироби просякають кам'яновугільним пеком, штучними смолами, тощо. Після просякнення їх повторно обпалюють.

У процесі обпалення відбувається коксування в'язучого, а заготовки, що обпалюються, спікаються в міцні моноліти. На перших стадіях обпалення із пресованих виробів видаляються леткі речовини, що присутні у в'язучому. Засипка, що щільно облягає заготовки, зберігає їх від деформації в період розм'якшення. Більша швидкість підйому температури тут не потрібна, адже вона може призвести до інтенсивного виділення летких речовин і розтріскування

виробу, що обпалюється. У подальшому в'яжуче коксується і коксова решітка, що утворюється при цьому, міцно зв'язує вуглецеві зерна наповнювача. Тривала витримка (близько 50 год.) за максимальної температури обпалення дозволяє отримати вугільні вироби з підвищеною щільністю.

Вугільні заготовки, що призначені для високотемпературної обробки завантажують в однофазну електричну піч опору, температура в якій піднімається до 2800 – 3300 К. Робочим опором в цій печі слугують самі вугільні заготовки, поміщені в пересипку із гранульованого коксу. При пропусканні електричного струму через робочий опір відбувається розігрів печі до високих температур.

Для отримання графіту високої чистоти на ряду з нагріванням використовують хлорно-фторну очистку. Графітовані вироби іноді зазнають додаткової обробки, а саме просякнення різноманітними речовинами та нанесення захисного покриття.

Із вищевказаного варто відмітити, що властивості вугільних графітованих виробів визначається особливостями вживаної вуглецевої сировини і технологічними факторами. Механічні і фізико-хімічні властивості вуглеграфітових матеріалів формують в процесі виробництва в залежності від призначення виробів [3].

В таблиці 1.1 наведені порівняльні характеристики різних видів вуглеграфітових матеріалів.

Таблиця 1.1 – Властивості вугільних та графітованих виробів

Показник	Електроди		Високодисперсний графіт	
	вугільні	графітовані	спосіб видавлювання	спосіб пресування в закриту форму
1	2	3	4	5
Максимальна	~10	~4	<0,1	<0,1

величина зерна, мм				
-----------------------	--	--	--	--

Продовження таблиці 1.1

1	2	3	4	5
Об'ємна густина, 10^3 кг/м^3	1,45 – 1,60	1,50 – 1,65	1,55 – 1,65	1,55 – 1,70
Пористість, %	25 – 30	23 – 30	20 – 27	20 – 25
Електроопір, 10^{-6} ом.м	40 – 65	6 – 13	6 – 13	8 – 20
Коефіцієнт лінійного розширення, 10^{-6} град^{-1} ($293^\circ - 473^\circ$ К)	4 – 5	1 – 3,5	1 – 3,5	1,5 – 5
Модуль Юнга, 108 Н/м^2	70 – 110	50 – 110	50 – 100	60 – 120
Межа міцності, 108 Н/м^2 а) на стиснення	250 – 500	150 – 300	200 – 400	250 – 600
б) на вигин	60 – 150	50 – 150	120 – 250	100 – 350
Теплопровідність, 4,18 $\text{кДж/м}\cdot\text{год}\cdot\text{град.}$	3,5 – 5	>100	>100	60 – 120

1.2. Конструкція печі графітування

На рисунку 1.2 схематично зображена завантажена піч графітування. Футеровка печі графітування виконується з шамотної цегли і складається з поду

1, торцевих стінок 2 та бічних стінок (на рисунку не вказані). Часто бічні стінки виконують розбірними, як правило із жаротривкого бетону. Стационарні бічні стінки та под, як правило, містять канали 3 для повітряного охолодження футеровки під час кампанії графітування.

У торцеві стінки печі вмонтовані струмопідвідні електроди 4, до яких із зовнішньої сторони під'єднуються пічні шини. Для запобігання окиснення електродів під час експлуатації, їх герметизують в місцях встановлення. Для цього у торцевих стінках передбачені порожнини 5, що щільно заповнені дрібнодисперсним графітом. Внутрішню поверхню торцевої стіни захищають за допомогою захисного екрану 6 із вуглецевих плит або вуглецевої маси товщиною 80..100мм. Верхню поверхню торцевих стінок захищають бетонними плитами 7.

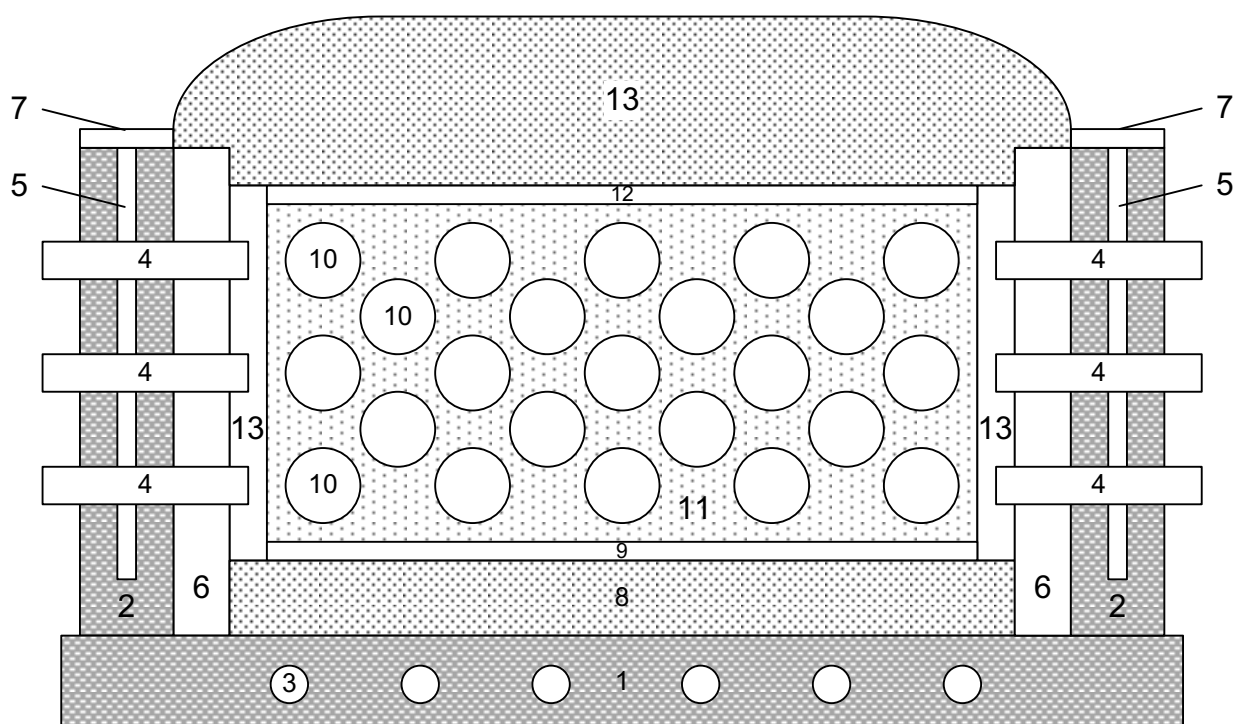


Рис.1.2 – Схематичне зображення завантаження печі Ачесона

Після невеликого ремонту піч графітування завантажують для наступної кампанії.

Для цього под печі вкривають шаром теплоізоляційної шихти 8 завтовшки 500..900мм. теплоізоляційна шихта складається з гранульованого

коксу фракції 0..10мм, піску та деревної тирси. Значну частину теплоізоляції може складати зворотна шихта.

Формування керну печі відбувається таким чином: бічні та торцеві стінки відгороджують щитами так, щоб між ними та стінками існував вільний простір, потім на шар теплоізоляції 8 насипають шар металургійного коксу 9 фракції 10..25мм. Після цього перпендикулярно до осі печі завантажуються перший ряд електродних заготовок 10, а простір між самими заготовками заповнюється керованою пересипкою 11. За таким самим принципом утворюють наступні шари із електродними заготовками. Зазвичай пересипку виготовляють із гранульованого металургійного коксу фракції 10..25мм, або дрібнішої, за графітування дрібної електродної продукції. Верх керну вкривають насипом із графітованого коксу 12. Коли kern сформовано, простір між щитами та бічними стінками заповнюють теплоізоляційною шихтою (на рисунку не вказано), а простір між щитами і торцевими стінками наповнюють графітованим коксом 13, після чого щити виймають. Наступним кроком є формування шару верхньої теплоізоляції 14 товщиною 500..800мм [3].

1.3. Піч графітування як об'єкт автоматизації

Графітування в печах Кастнер, в порівнянні з графітуванням в печах Ачесона, більш вигідно як з точки зору витрати матеріалів, тривалості процесу, так і з точки зору більш рівномірного графітування заготовок по їх перетину і довжині. Однак, організація процесу графітування в печах Кастнер вимагає більш потужного електроживлення обладнання і підвищеної якості вихідного матеріалу заготовок в порівнянні з печами Ачесона. Формування робочого простору в цих печах ускладнено необхідністю створення якісного контакту між торцями заготовок, а самі печі мають більш складну конструкцію - вузол притиску заготовок, охолодження рухомого струмопідводу і т.д. Тому історично більш широке поширення отримали печі Ачесона, які представляють собою модернізовані печі для карбонізації. Саме печі такого типу використовуються на більшості електродних заводів СНД, і зокрема на ПАТ «Укрграфіт».

Графітизація вуглецевої речовини – це процес впорядкування атомів вуглецю в структурі графіту під впливом високих температур. На процес графітування впливає ряд факторів, особливо індивідуальні особливості вуглецевих матеріалів, газове середовище, передісторія термічної обробки, тривалість процесу, температура тощо. Дії декотрих із них на процес графітування до цих пір не цілком ясна. Однак, як показують численні дослідження, головним фактором, який визначає графітування вуглецевої речовини, застосовної для виготовлення електродів, є кінцева температура. При чому, для отримання якісного штучного графіту кінцева температура обробки повинна бути не нижче 2200 – 2800°C.

Процес графітування електродних заготовок в промисловості здійснюється в електричних печах опору, в яких сам матеріал, що завантажується у піч, слугує активним електричним опором у загальному ланцюгу агрегату графітування: Джерело живлення – коротка мережа – піч. Матеріал, що завантажується, розігрівається у результаті безпосереднього протікання по ньому електричного струму, через це печі графітування відносять до категорії печей прямого нагріву. Для аналізу процесу нагріву власне електродних заготовок, або, точніше, принципів формування температурного поля у об'ємі печі графітування, розглянемо структуру завантаження печі як електричного опору. На рисунку 1.2 наведено повздовжній переріз печі графітування, із якого видно, що завантаження печі являє собою чергування рядів заготовок та вуглецевої шихти (пересипки), що оточена з усіх боків шаром теплоізоляційної шихти. До того ж питомий електричний опір (ПЕО) пересипки значно вище (приблизно в 50 – 100 разів), ніж ПЕО заготовок. В результаті цього все джоулеві тепло виділяється у шарі пересипки, а нагрів електродних заготовок здійснюється теплопровідністю та тепловим випромінюванням від частинок пересипки. Через це нагрів власне електродних заготовок в печах графітування є опосередкованим.

При опосередкованому нагріві будь-якого твердого тіла (в даному випадку обпалених вуглецевих заготовок) у ньому виникають температурні

градієнти, а отже, термічні напруження. Коли швидкість зміни температури велика, варто враховувати і динамічні ефекти, обумовлені змішуванням частинок тіла.

Експериментально цю задачу виконати складно так як дослідження високотемпературних полів у вуглецевому середовищі є досить складним і, що найголовніше, отримана інформація обмежена. Для вирішення поставленої задачі розроблена математична модель промислового процесу графітування в електричних печах опору.

Для підвищення якості отримуваних виробів обладнання забезпечене екстремальним регулятором, вхід якого підключений через суматор до виходів других елементів порівняння, а виходи — до коректорів. У процесі роботи пристрою екстремальний регулятор задає коефіцієнти передачі коректорів, підбираючи їх такими, щоб звести до мінімуму поточні відхилення параметрів розплаву від заданих. Зменшення відхилень дозволяє підвищити точність управління режимом екструзії і якість виробів.

При опосередкованому нагріві будь-якого твердого тіла (в даному випадку обпалених вуглецевих заготовок) у ньому виникають температурні градієнти, а отже, термічні напруження. Коли швидкість зміни температури велика, варто враховувати і динамічні ефекти, обумовлені змішуванням частинок тіла.

Щоб вирішити цю задачу необхідно передусім знати основні закономірності формування температурних полів, як в тілі заготовки, так і в об'ємі печі в цілому. Через це розглянемо характеристику печі графітування, як теплового об'єкта.

Наука про теплопередачу розглядає такий об'єкт, як складене анізотропне тіло, тепло- та електрофізичні властивості які залежать від його температури, а сам процес графітування в електричних печах — як задачу нестационарної теплопровідності із внутрішнім джерелом тепла за змінних граничних умовах. Інакше кажучи, в печі графітування тепловиділення та теплообмін відбувається у тісному взаємозв'язку, і через це не можна, наприклад, розглядати формування

температурного поля у заготовках та пересипці незалежно від властивостей теплоізоляції або вирішувати задачу нагріву теплоізоляції не враховуючи конкретний режим вводу тощо. Сукупність усіх цих процесів, що відбуваються у робочому просторі печі за рахунок тепловиділення, і називаються тепловою роботою.

Теплову роботу визначають на корисну, котра являє собою передачу тепла електродним виробом, що нагріває і втраченою, що включає всі інші види споживання тепла.

Через це основною задачею теплового розрахунку є визначення корисної потужності, теплових втрат і розрахунок нестационарних теплових полів у заготовках, що нагріваються, пересипці, теплоізоляції та конструктивних елементах печі.

Аналітичне описання процесу теплопровідності у нерухомому анізотропному середовищі базується на законах збереження енергії і Фур'є, і призводить до диференціальних рівнянь теплопровідності із внутрішнім джерелом тепла, котрі зв'язують часову та просторову зміну температури. У дані задачі джерелом внутрішнього тепла є джоулеві енергія. Через це необхідно також вирішити задачу електромагнітного поля печі.

Рівняння процесу графітування здійснюється по електричним параметрам: потужності, струму, напрузі. Таким чином, щоб керувати процесом графітування або, що теж саме, керувати температурним полем печі, необхідно встановлювати взаємозв'язок між електричними характеристиками процесу і формуванням температурного поля.

Сучасні світові тенденції у виробництві сталі пов'язані передусім з переходом галузі на використання дугових сталеплавильних печей високої та надвисокої потужності (70 .. 180 МВА), що вимагає постійного нарощування обсягів виробництва графітованих електродів великого діаметра, які припускають підвищену густину струму та характеризуються високою механічною міцністю і термостійкістю.

Виробництво графітової продукції складне, багатостадійне та дуже енергоємне. При цьому на заключну стадію – графітування – припадає до 80 % загальних енерговитрат [4]. В умовах постійного зростання вартості електроенергії зменшення енерговитрат на графітування при забезпеченні високої якості кінцевої продукції (підвищення ефективності процесу) є актуальною науковою і технічно графітування великогабаритної продукції, до якої належать електроди дугових сталеплавильних печей, подові блоки алюмінієвих електролізерів тощо, як правило, здійснюють в електричних печах опору по технології Ачесона. Печі Ачесона характеризуються істотною нерівномірністю розподілу температурного поля та нестабільністю електро- та теплофізичних властивостей матеріалів завантаження.

Підвищити ефективність графітування можна інтенсифікацією процесу. При швидкому нагріванні печі збільшується нерівномірність температурного поля і, як наслідок, збільшуються термонапруження у виробках. Термічна міцність виробів виступає основним чинником, що обмежує інтенсифікацію процесу. При цьому опосередкована оцінка термопружного стану виробів призводить до перевитрат електроенергії та знижує ефективність виробництва в цілому. Іншим чинником перевитрат електроенергії є неточне визначення необхідної тривалості кампанії графітування.

Задачу підвищення ефективності печей Ачесона можна розв'язати шляхом впровадження системи керування, що реалізує оптимальне ведення процесу графітування на основі аналізу термопружного стану виробів та ступеня їх графітування.

Неможливість прямого контролю властивостей електродних виробів під час кампанії графітування, а також відсутність промислових методів вимірювання температурного поля печі Ачесона у всьому діапазоні температур обумовлює той факт, що досьогодні жодне графітове виробництво в Україні та країнах СНД не обладнано системою керування, яка б забезпечила оптимальну динаміку підведення потужності в реальному часі.

2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ У ПЕЧІ АЧЕСОНА

2.1. Існуючі математичні моделі процесів у печі Ачесона

В роботах [5, 6] розглянуто моделювання струморозподілу в системі під графітування - шинопакет. Модель представлено у вигляді еквівалентної схеми заміщення. Методику призначено для розрахунку індуктивності керна та короткої мережі при проектуванні ошиновки печей графітування. Для оцінки енергетичного стану модель непридатна.

В роботі [7] досліджено явище розтікання струму у вуглецевих виробках. Модель електричного поля окремого зразка в печі постійного струму отримано в аналітичній формі методом аналогії (з електростатичним полем). Адекватність моделі перевірена експериментально та зроблено висновок про відсутність впливу неоднорідності вуглецевого матеріалу зразка на характер розтікання струму.

В роботі [8] запропоновано в спрощеній аналітичній формі визначати розподіл густини струму в міжрядовій пересипці. За результатами моделювання зроблено висновок про нерівномірність виділення теплоти на поверхні заготовок, що зумовлює неодночасне протікання графітування і, як наслідок, неоднорідність властивостей в об'ємі заготовок.

В роботі [9] зроблена спроба оцінки перепадів температур в керні печі на заключній стадії кампанії графітування, яку автори вважають стаціонарною (стале розподілення температури в керні). Для цього введено спрощуючі припущення про однорідність електро- та теплофізичних властивостей матеріалів завантаження по всьому об'єму печі, однорідність розподілення в керні внутрішнього джерела теплоти, та про циліндричну форму керна (діаметр циліндра дорівнює ширині керна). Модель температурного поля представлено у вигляді стаціонарного одновимірного рівняння теплопровідності і розв'язується в аналітичній формі. Модель можна застосовувати тільки для орієнтовного розрахунку рівноважного перепаду температур керна, що в промисловій практиці не має місця.

В роботі [10] досліджено температурне поле електродної заготовки в керні печі графітування. Модель поля представлено рівнянням теплопровідності з внутрішнім джерелом теплоти та крайовими умовами. Розв'язання задачі здійснено на фізичній моделі (експериментальній печі) за допомогою критеріїв подібності методом інтегральної аналогії. Розподіл густини струму в перерізі керна визначається за допомогою спрощеної аналітичної залежності.

В роботі [11] розглянуто методику моделювання температурного стану печі графітування на аналогових обчислювальних машинах. Просторова динамічна модель базується на розв'язанні електро-термічної задачі. В об'ємі керна печі виділено симетричний розрахунковий фрагмент (РФ) спрощеної геометрії з гомогенними властивостями. Електромагнітне поле РФ описано системою стаціонарних диференціальних рівнянь Лапласа. Температурне поле РФ описано системою нелінійних рівнянь теплопровідності з крайовими умовами 1-го та 3-го роду. Задача розв'язується в сітковій формі методом контурної апроксимації з використанням електротеплової аналогії. Модель дозволяє визначати приблизне температурне поле заготовок з перерізом прямокутної форми в печі графітування і, на думку авторів, може застосовуватися при розрахунках футеровок печей графітування та для оптимізації товщини шару бокової теплоізоляції.

В роботі [12] на основі диференціального рівняння енергії для одновимірного теплового потоку запропоновано модель нагрівання заготовок до температури 1500°C . Модель розв'язується в скінченних різницях сітковим методом. Густина тепловиділення в області пересипки вважається відомою, а в області заготовки внутрішнє джерело теплоти відсутнє. Модель призначено для орієнтовних розрахунків градієнту температур в заготовках.

Через значні спрощення розглянуті моделі мають низьку точність відтворення реальних процесів і можуть використовуватися лише для наближеного аналізу температурних полів печі графітування.

В роботі [13] розглянуто спосіб електромоделювання нестационарного температурного поля твердого тіла з крайовими умовами 3-го роду, заснований

на методі елементарних балансів. Тверде тіло апроксимують множиною елементарних паралелепіпедів, для кожного з яких складається рівняння балансу енергії. Розв'язання моделі здійснюють за допомогою електричної схеми, зібраної у вигляді сітки відповідно до закону Кірхгофа так, щоб відтворити в кожному з вузлів аналог теплового балансу для відповідного елементарного паралелепіпеда.

Ідеї метода елементарних балансів застосовані при моделюванні температурного поля печі графітування в роботах [14, 15]. Модель тут будується на основі аналізу симетричного РФ, що представлено складеним твердим тілом, в якому електро- та теплофізичні властивості структурних елементів мають суттєву залежність від температури. Температурне поле фрагменту описується системою диференціальних рівнянь теплопровідності з внутрішнім джерелом теплоти, початковими та крайовими умовами. Для кращої збіжності метода та зменшення обсягу розрахунків, циліндричну форму електродних заготовок замінено на прямокутну із збереженням оригінального об'єму заготовок. При цьому зміна відстані між заготовками враховується поправочними коефіцієнтами, визначеними з умови збереження маси заготовок та пересипки. Для розв'язання задачі розподілення густини струму в сітці РФ прийнята електротехнічна схема заміщення, що розраховується по формальним правилам для всіх горизонтальних (паралельних) рядів заготовок на кожному часовому кроці.

Описану модель вдосконалено в роботі [16]. Для зниження похибки відтворення експериментальних даних в області високих температур (вище 2800°C) ефективна теплоємність вуглецевих матеріалів завантаження печі замінена на дійсну, а енергію, що витрачається на сублімацію вуглецю, враховано в елементарних балансах тільки як стік теплоти, розрахований за різницею ефективною та дійсною теплоємності (при 3000°C ефективне значення перевищує дійсне в 1.2 .. 1.5 разів).

Розглянуті моделі призначено переважно для дослідження теплової роботи печей графітування при існуючих регламентах підведення потужності.

Оскільки форма електродних заготовок в моделях суттєво відрізняється від форми їх фізичного прототипу, важко робити висновки про дійсне значення градієнту температур в об'ємі заготовок, тим більше, неможливо застосовувати моделі при аналізі термопружного стану виробів. Крім того, моделювання засновано на сіткових алгоритмах [17, 18], які не дозволяють якісно враховувати крайові умови на поверхнях складної форми, а значний час розрахунку обмежує використання моделей в системах керування.

Загальні підходи до моделювання термопружних станів виробів з вуглецево-графітових матеріалів наведено в [19, 20]. Тут розглянуто методики розрахунку термонапружень у вільних циліндрах із анізотропного вуглецевого матеріалу з врахуванням залежності фізико-механічних властивостей матеріалу від температури. Модель включає рівняння рівноваги, рівняння сумісної деформації та рівняння закону Гука. Система рівнянь розв'язується методом Гальоркіна при однорідних крайових умовах. Наведені методики незастосовні для аналізу термопружного стану електродних заготовок в печі графітування, оскільки стосується порожнистих одношарових та тришарових циліндрів.

В роботі [21] розглянута можливість розв'язання теплової задачі та задачі міцності для одиничної заготовки, що рівномірно нагрівається. При цьому прийнято припущення про симетричний відносно осі і однаковий за довжиною розподіл температури в заготовці (тобто тепловий потік в заготовці спрямовано виключно в радіальному напрямку та напружений стан заготовки є пласко-деформованим). Розрахунок нестационарного температурного поля в заготовці здійснено методом прогонки для неявної скінченно-різницевої схеми [22, 23]. На підставі розрахунку напружень в середньому перерізі заготовки автори роблять висновок про можливість скорочення процесу графітування більш, ніж у 10 разів при рівномірному розігріві керна. Такий характер розігріву в печах Ачесона неможливий і помилковість висновків авторів, що базується на нереальних припущеннях, доведена численними експериментами [24-26]. Незважаючи на це, цінність роботи полягає в тому, що результати моделювання підтвердили принципову можливість суттєвої інтенсифікації процесу

графітування на основі аналізу термопружного стану заготовок в печах графітування.

Для адекватного відображення процесів, що відбуваються в печі, за допомогою моделей необхідно знати електро- та теплофізичні властивості матеріалів завантаження як функції температури.

Дослідженню властивостей вуглецевих матеріалів присвячено чимало публікацій, наприклад [27-33], але тільки в деяких, наприклад [34-37], розглянуто залежність властивостей від температури в широкому діапазоні. При цьому практично відсутні дослідження властивостей сипких матеріалів на основі гранульованого вуглецю при високих температурах (до 3300°C). В роботі [38] розглянуто зміну коефіцієнту теплопровідності різних типів теплоізоляційної шихти та пересипки тільки до температури 1600°C. В роботі [39] досліджено тепло- та електропровідність пересипки на основі гранульованого коксу до температури 2200°C. В роботі [40] приблизні температурні залежності властивостей матеріалів завантаження для всього температурного діапазону графітування отримано на основі узагальнення літературних даних та шляхом розв'язання зворотної задачі теплопровідності при відомому температурному полі з подальшою екстраполяцією в область високих температур.

В роботі [41] зазначено, що питомий опір пересипки від кампанії до кампанії змінюється непередбачувано (навіть при однаковій рецептурі пересипки та однаковому регламенті розкид електричного опору печі для того ж періоду різних кампаній може сягати 15 %). Тому дані, отримані з літературних джерел, можуть розглядатися тільки як перше наближення при ідентифікації об'єкту і потребують подальшої адаптації. Наприклад, в роботі [41], для досягнення збігу результатів моделювання з експериментальними даними здійснюється ручна корекція ПЕО пересипки, а в роботі [42], крім корекції ПЕО пересипки, здійснюється також корекція коефіцієнту теплопровідності теплоізоляційної шихти.

2.2. Способи виміру температури в печі графітування

Основним параметром, що визначає перебіг процесу графітування, є температура. Швидкість підвищення температури визначає ефективність процесу: якість виробів та витрати електроенергії. Максимально досягнутий рівень температур в «найхолодніших» точках виробів визначає тривалість кампанії. А максимально досягнутий рівень температур на межі теплоізоляція – футеровка впливає на строк служби будівельних конструкцій печі. Визначення температури у внутрішньому просторі печі графітування – це надзвичайно складна технічна задача. Серед чинників, що ускладнюють контроль, головними є високі рівні температур (в деяких місцях керна температура перевищує 3000°C), значні геометричні розміри об'єкту (складність розташування датчиків в контрольних точках при прямих методах вимірювання), агресивна атмосфера внутрішнього простору (вуглець при високих температурах виступає активним відновником, наявність газів з великим вмістом сірки та кремнію тощо), постійна усадка об'єму завантаження, вплив потужних електромагнітних полів, забрудненість повітря вуглецевим пилом тощо.

Розглянемо можливі способи, якими можна виміряти температуру під час кампанії графітування, та їх потенції у використанні.

Для безпосереднього вимірювання температури в печах графітування застосовують термопари вольфрам-ренієві (ТВР) і хромель-алюмелеві (ТХА), графітові дилатометри, оптичні пірометри та ін. Вольфрам-ренієві і хромель-алюмелеві термозонди представляють собою керамічний чохол з окису алюмінію або окису магнію, всередині якого знаходяться термопари, захищені керамічною високотемпературної соломкою (окис алюмінію, берилію або магнію). Порожнечу в керамічному чохлі заповнюють порошком окису магнію або нітриду бору. Рівень температур, який дозволяють виміряти вольфрам-ренієві термозонди в інертному середовищі або у вакуумі, може досягати $2500\text{--}2800^{\circ}\text{C}$, однак, при вимірах в печі цей рівень внаслідок присутності агресивно-відновлювальної середовища істотно знижується і зазвичай не перевищує рівень 2000°C . Рівень температур, який дозволяють виміряти в печі графітування

хромель-алюмелеві термозонди знаходяться в діапазоні 1100-1200 ° С. Істотно ускладнює вимірювання температури з допомогою термозондів із захисним керамічним чохлам те, що під час проведення кампанії відбувається нерівномірна усадка матеріалів керна, внаслідок чого керамічні чохла, що володіють низькою механічною міцністю, часто руйнуються і термопари виходять з ладу. Вимірювання температур за допомогою графітових дилатометрів засновано на порівнянні лінійного подовження дилатометричні тіла (спеціальні марки пірографіта) і графітованої стержня, що дозволяє надійно вимірювати температуру в інтервалі 800-3000 ° С. Однак використання цього способу вимірювання ускладнене через високо рівня просадок матеріалів в керна і надзвичайно низькою механічної міцності дилатометричні тіла.

Перераховані засоби виміру температури відносяться до контактних датчиків, а до безконтактних - оптичні пірометри, ермопоінти і тепловізори. За допомогою термопоінтов і тепловізорів визначаються в основному температури зовнішніх поверхонь - кладки печі, струмопідводів. Оптичний пірометр використовується для високотемпературних вимірювань (понад 900 ° С) температур керна печі. Для виконання вимірів за допомогою оптичного пірометра при завантаженні печі в досліджувану зону встановлюється графітовий блок з отвором, є точкою візира (візирної трубкою). Щоб забезпечити реєстрацію температури в будь-який момент часу і збільшити точність вимірювань, необхідно видалити димну хмару хоча б на час вимірювання, що досягається наступними прийомами:

- продувкою візирної трубки перед кожним виміром;
- установленням у кінці візирної трубки вертикально до неї другої трубки, яка виходить за межі печі і завдяки якій створюється природна тяга, в результаті чого в візирної трубці виходить відносно чиста атмосфера;
- створенням підвищеного тиску в візирної трубці, що не дозволяє проникати газам в трубку. Тиск створюється інертним газом. Оптичний пірометр монтується або разом з трубкою, або через кутовий відбивач.

Основним недоліком вимірювання температури за допомогою оптичного пірометра є суттєва похибка вимірюваних температур через впливу масивного високо теплопровідні графітового блоку, зниження рівня температур під час продування холодним інертним газом і т.д. Крім перерахованих способів вимірювання для дослідження температури керна також застосовуються оригінальні методи вимірів, наприклад метод свідків. Метод «свідків» - передбачає використання в якості термоіндикатора важко графітувальний матеріал - скловуглець (СВ), що характеризується лінійним зміною фізичних характеристик при високих температурах. В якості реєстрованого параметра пропонується використовувати або діамагнітну сприйнятливості СВ або ТЕДС. До недоліку цього методу відноситься забруднення в реальних умовах матеріалу - свідка, що призводить до зміни його фізичних властивостей і спотворення результатів вимірювань. Також оригінальний спосіб вимірювання температури в печі графітування є: виготовляється тепловід у вигляді стрижня, один кінець якого знаходиться в контакті з об'єктом вимірювання температури, а другий підтримується при постійній температурі примусовим охолодженням. Бічна поверхня теплоізолювана. При практичному користуванні вимірюється температура в проміжній точці стрижня і по номограмі визначається температура гарячого кінця (Граничне її значення дорівнює 2400°C). З наведених вище даних випливає, що в промислових умовах проблема достовірного прямого виміру температур, що характеризують закінчення процесу графітування (понад 2400°C) не вирішена остаточно. Більш того, розглянуті методи оцінки температур керна досить обмежені через відносно невеликого числа контрольованих точок, великої складності експерименту і високою вартості експериментального обладнання. Тому для оцінки температурних полів в печах графітування найбільш прийнятним є використання методів математичного моделювання.

Визначити температуру в будь-якій точці печі графітування можна шляхом розрахунку температурного поля за допомогою математичної моделі, що

враховує геометричні розміри об'єкту, теплофізичні властивості матеріалів та поточні витрати енергії на графітування[3].

2.3. Математичне моделювання фізичних полів під час графітування заготовок у печі Ачесона

Перед тим, як формулювати математичну постановку задачі теплоелектричних полів печі Ачесона необхідно спочатку визначити, які фізичні процеси, що відбуваються при графітуванні електродних виробів, впливають на формування цих полів, тобто сформулювати фізичну модель процесу графітування заготовок у печі непрямого нагріву. Ґрунтуючись на даних літературних джерел та досвіду експериментально-теоретичних досліджень авторів фізична модель процесу графітування в печі Ачесона може бути сформульована наступним чином.

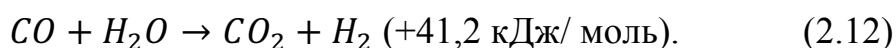
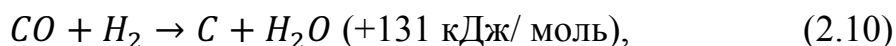
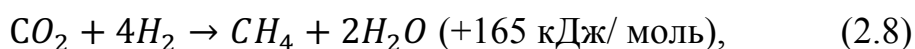
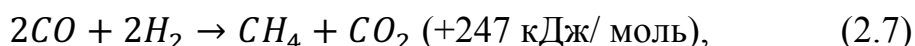
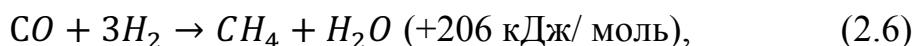
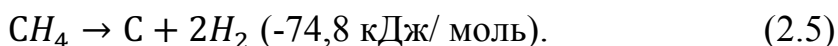
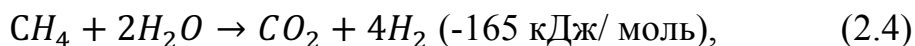
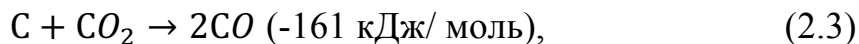
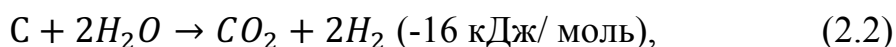
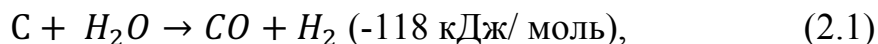
Робочий простір печі Ачесона представляє собою витягнутий паралелепіпед з відкритим верхом, над яким встановлений дзвіноподібний газозбірник, з'єднаний з димовідсмокчувачем. Робочий простір печі заповнюється заготовками, які засипаються електропровідною та теплоізоляційною пересипкою. Електропровідна і теплоізоляційна пересипка представляє собою вологий сипучий матеріал і відрізняється між собою ефективним діаметром гранул. Простір між гранулами заповнений димовими газами. Таким чином, можна сказати, що простір між заготовками і стінками печі заповнений двофазним середовищем. Через піч пропускається електричний струм. Електричне поле печі вважається безвихровим з відсутністю термоелектричних ефектів. Нагрів заготовок і теплоізоляційної пересипки відбувається непрямым шляхом за рахунок джоулевої теплоти, що виділяється в електропровідній пересипці і хімічних окисних реакцій. Теплота, що виділяється в печі, також витрачається на нагрів оточуючих елементів, відновні хімічні реакції і втрати в навколишнє середовище. В процесі нагріву печі відбувається інтенсивне випаровування вологи, що призводить до стрибкоподібної зміни фізичних властивостей пересипки.

Випаровування вологи в процесі розігріву печі починається в центральній зоні кернової пересипки. Пара, що утворилася, розширюється і викликає переміщення незв'язаної вологи, що не випарувалась, в периферійні холодні шари кернової пересипки, де відбувається її конденсація. Подальше випаровування вологи з кернової пересипки супроводжується проникненням пари в теплоізоляційну шихту, виштовхуванням незв'язаної вологи до стінок і підлоги печі, конденсацією пари в «холодних» шарах теплоізоляційної шихти і частково його евакуацією в навколишнє середовище. Сконденсована в теплоізоляційній шихті вода під дією гравітаційних сил опускається донизу, тим самим підвищуючи концентрацію вологи в подинній теплоізоляції. При подальшому нагріві печі відбувається випаровування вологи з теплоізоляційної шихти з механізмом переміщення пари і вологи аналогічно описаному вище. Варто відзначити, що при досягненні певної концентрації незв'язаної вологи в сипучих слоях пересипки може спостерігатися її переміщення в зони з меншою концентрацією вологи (більш нагріті, центральні зони) під дією капілярних сил.

Утворена при випаровуванні волога верхніх шарів подинної теплоізоляції і прикернових шарів бокової теплоізоляції пара, практично в повному об'ємі, проходить через нагріту кернову пересипку, яка повністю складається з вуглецевого матеріалу. Внаслідок чого, при високих температурах кернової пересипки (понад 650°C) відбувається конверсія водяної пари, яка супроводжується газифікацією кернового коксу зі значним поглинанням теплоти. Основні хімічні реакції газифікації кернової коксової пересипки і їх тепловий ефект (при температурі кернової пересипки близько 800°C) наведені в рівняннях (2.1-2.5).

Продукти реакції газифікації – оксид вуглецю і водень, частково евакуюються в навколишнє середовище, і частково, проходячи через «холодні» шари теплоізоляції, вступають в екзотермічні реакції з утворенням води і метану. Основні хімічні реакції, що відбуваються в зоні конверсії оксиду вуглецю та водню і їх тепловий ефект (при температурі теплоізоляції близько 200°C) наведені в рівняннях (2.6 – 2.12). Описана схема механізму конверсії водяної

пари, оксиду вуглецю та водню в печі Ачесона відображена відповідними формулами.



Утворена, в результаті конверсії окису вуглецю і водню водяна пара, конденсується і під дією гравітаційних сил опускається в подинну теплоізоляцію, де знову випаровується.

Таким чином, завдяки механізму переміщення вологи і пари і механізму газифікації коксової пересипки відбувається циркуляція води і пари в об'ємі печі до повної її евакуації в оточуюче середовище.

З наведених рівнянь хімічних реакцій (2.1) – (2.12) видно, що більшість з цих реакцій можуть бути як прямими, так і зворотніми. Напрямок наведених хімічних реакцій залежить від температури і визначається константами рівноваги. За повного випаровування вологи з верхніх шарів подинної, прикернової і бокової теплоізоляції процес газифікації кернової коксової пересипки завершується.

При досягненні у шарах теплоізоляційної шихти, що вміщують дерев'яну тирсу, температур вище 250°C відбувається процес піролізу, що супроводжується виділенням теплоти. Подальший нагрів печі супроводжується

хімічними реакціями відновлення оксидів металів, що знаходяться в теплоізоляційній шихті. Реакції (2.10) – (2.12) є ендотермічними і відбуваються при високих рівнях температур теплоізоляційної шихти. У зв'язку з високим вмістом піску в теплоізоляційній шихті основною реакцією відновлення оксидів металів мають місце реакції, що відбуваються при температурах більше ніж 1670°C.

У відповідності з вище викладеною фізичною моделлю розробимо математичну модель фізичних полів печі Ачесона в процесі графітування електродних заготовок.

Розроблена математична модель ґрунтується на системі рівнянь тепло- та електропровідності, механічного руху і рівноваги на базі моделі Дракера-Прагера, розробленої для сипких матеріалів, і описує теплоелектричний і механічний стан печі з врахуванням контактних взаємодій теплоелектричної і механічної природи на межі сипкий матеріал – тверде тіло.

Відмінною особливістю розробленої математичної моделі є врахування вищезазначених механізмів переносу вологи і пари, переносу теплоти за рахунок реакцій газифікації кривої коксової пересипки, конверсії окису вуглецю і водню в теплоізоляційній шихті, а також теплоти піролізу деревини, що відбуваються в печі.

Систему диференціальних рівнянь фізичних полів печі Ачесона з використанням механічної моделі сипкого середовища Дракера-Прагера можна записати наступного вигляду:

$$\begin{cases} \frac{\partial H}{\partial \tau} = \nabla \cdot [\lambda_{eff}(p, T) \nabla T(X)] + \chi(p, T) |\nabla U(X)|^2 + q_{V, chem} + q_{V, devapor}; \\ \nabla \cdot [\chi(p, T) \nabla U(X)] = 0; \\ \nabla \cdot \dot{\sigma} \cdot \dot{f} = 0, \end{cases}$$

де $H = \int_{T_{ref}}^T c_P^{eff}(T) \rho(T) dT$ – ентальпія, в якій враховується теплота випаровування чи конденсації вологи у вологовмісних матеріалах печі, Дж/м³; $X(x, y, z) \in R^3$ – декартова система координат, м;

$$c_p^{eff}(T) = \begin{cases} c_{p-0}^*(T), T < T_{eqL}; \\ \frac{c_{p-0}^*(T_{eqL}) + c_{p+0}^*(T_{eqR})}{2} + \frac{L_{fm}}{\Delta T}, T_{eqL} \leq T \leq T_{eqR}; \\ c_{p+0}(T), T > T_{eqR}, \end{cases} \quad (2.13)$$

c_p^{eff} – ефективна масова ізобарна теплоємність (2.13), що враховує в матеріалах, що містять вологу теплоємність води і теплоту її пароутворення, а в матеріалах, що підлягають графітування, теплоту сублімації графіту при високих температурах, Дж/ (кг·К);

$$T_{eqL} = T_{eq} - \frac{\Delta T}{2}; T_{eqR} = T_{eq} + \frac{\Delta T}{2}; T_{eq} \quad (2.14)$$

T_{eqL} – рівноважна температура випаровування, сублімації (фазового переходу першого роду) чи хімічної реакції, К; ΔT – температурний інтервал згладжування, К; c_{p-0}^* – масова теплоємність матеріалу з врахуванням його вологовмісту, Дж/ (кг·К);

$$L_{fm} = \begin{cases} \alpha_b L_{fv} - \text{при випаровуванні води;} \\ (1 - \alpha_b) L_{fc} - \text{при сублімації графіту;} \end{cases} \quad (2.15)$$

L_{fm} – масова теплота при фазових переходах першого роду, Дж/ кг; α_b – масова частка води у вологому матеріалі; L_{fv}, L_{fc} – масова теплота фазового переходу при випаровування води і сублімації графіту, відповідно, Дж/ кг; q_{Vchem} – густина внутрішнього джерела теплоти, пов'язана з хімічними реакціями 2.10-2.15, 2.1-2.12 і піролізом деревини, Вт/ м³; $q_{V,devapor}$ – густина внутрішнього джерела теплоти, що враховує тепломасоперенос водяної пари і її конденсацію, Вт/ м³; $\dot{\sigma}, \dot{f}$ – похідні по часу тензора напруги другого рангу і вектору об'ємних сил, пов'язаного з графітацією, відповідно; τ – час, с.

Ентальпія, що враховує вологовміст теплоізоляції, керованої пересипки і теплоти сублімації вуглецевовмісних матеріалів печі Ачесона з врахуванням вищенаведеної залежності $c_p^{eff}(T)$, визначається наступним чином:

$$H = \int_0^{T_{eqL}} c_{p-0}^*(T) \rho_{-0}^*(T) dT + \frac{[c_{p-0}^*(T_{eqL}) \rho_{-0}^*(T_{eqL}) + c_{p+0}^*(T_{eqR}) \rho_{+0}^*(T_{eqR})]}{2} \Delta T + \\ + L_{fv} + \int_{T_{eqR}}^{T > T_{eqR}} c_{p+0}(T) \rho_{+0}(T) dT,$$

де c_{p-0}^*, ρ_{-0}^* і c_{p+0}, ρ_{+0} - властивості матеріалів по обидві сторони від температурного фазового переходу (з врахуванням воловмісту), відповідно;

$$L_{fv} = \begin{cases} \alpha_B \rho_B L_{fB} - \text{при випаровуванні води;} \\ (1 - \alpha_B) \rho_G L_{fG} - \text{при сублімації графіту;} \end{cases} - \text{об'ємна теплота при}$$

фазових переходах першого роду, Дж/м³; ρ_B, ρ_G - густина води і графіту, відповідно, кг/м³.

Ефективна теплопровідність матеріалів з врахуванням вологовмісту визначається залежністю

$$\lambda_{eff}(p, T) = \begin{cases} \lambda_{-0}(p, T), T < T_{eqL}; \\ \lambda_{-0}(p, T_{eqL}) + \frac{\lambda_{+0}(p, T_{eqR}) - \lambda_{-0}(p, T_{eqL})}{\Delta T} T, T_{eqL} < T < T_{eqR}; \\ \lambda_{+0}(p, T), T > T_{eqR}, \end{cases}$$

де λ_{-0} і λ_{+0} - теплопровідність матеріалу по обидві сторони від температури фазового переходу, відповідно, Вт/(м·К).

Густина внутрішнього джерела теплоти, пов'язана з хімічними реакціями при газифікації коксової кривої пересипки, описується співвідношенням

$$q_{Vchem} = \frac{Q_{r.g} m_{vapor}}{V_{gasific} \tau_{gasific}} \quad (2.16)$$

де $Q_{r.g}$ - тепловий ефект реакції газифікації, Дж/кг; m_{vapor} - маса пари, що вступила у реакцію газифікації, кг; $V_{gasific}$ - об'єм зони газифікації, м³; $\tau_{gasific}$ - тривалість процесу газифікації, с.

За умови, що основною реакцією процесу газифікації є реакція «водяного газу» (2.1), величини $Q_{r.g}$ визначається по (2.17), а m_{vapor} - по (2.18).

$$Q_{r.g} = (-7165100 - 718,25T + 0,1162T^2 + 0,00007903T^3), \quad (2.17)$$

$$m_{vapor} = 1,5m_c = 1,5 \min \left(\frac{ke^{-\frac{E}{RT}} m_{cgasif} \tau_{gasific}}{0,6667 m_{all.vapor}} \right), \quad (2.18)$$

де T - середньомасова температура вуглецю в зоні газифікації, К; m_c - маса вуглецю, що вступив у реакцію газифікації, кг; k - передекспоненційний множник, с⁻¹; E - енергія активації, Дж/моль; R - газова стала, Дж/(моль·К);

m_{Cgasif} – маса вуглецю в зоні газифікації, кг; $m_{all.vapor}$ – маса водяної пари, що пройшла через зону газифікації, кг.

Для процесу газифікації керованої коксової пересипки можна прийняти, що $k = 833,33 \text{ с}^{-1}$, $E = 133,2 \text{ Дж/ моль}$, а m_{Cgasif} як масу керованої коксової пересипки з температурою більше 600°C .

Густина внутрішнього джерела теплоти, пов'язана з хімічними реакціями конверсії окису вуглецю і водню в теплоізоляційній шихті(в основному реакція гідрування оксиду вуглецю (2.6), у спрощеному вигляді можна визначити як

$$q_{V,chem} = \frac{Q_{CH_4} m_{CO}}{V_{convers} \tau_{convers}} k_{evac.g} = \frac{Q_{CH_4} \frac{1}{3} \cdot \frac{28 \cdot 2}{2 \cdot 18} m_{vapor}}{V_{convers} \tau_{convers}} k_{evac.g} \approx \frac{Q_{r.g} m_{vapor}}{V_{convers} \tau_{convers}} k_{evac.g} \quad (2.19)$$

де $k_{evac.g}$ - коефіцієнт. Що враховує евакуацію окису вуглецю і водню в навколишнє середовище; Q_{CH_4} - тепловий ефект реакції (2.6), Дж/ кг; m_{CO} - маса окису вуглецю, що вступив у реакцію утворення метану (6); $V_{convers}$ - об'єм зони конверсії окису вуглецю і водню в теплоізоляції, м^3 ; $\tau_{convers}$ - тривалість процесу конверсії окису вуглецю і водню в теплоізоляційній шихті, с.

Густина внутрішнього джерела теплоти, що враховує теплоту піrolізу деревини, визначається наступною залежністю

$$q_{Vchem} = \frac{Q_{p.w} m_w}{V_w \tau_w} \quad (2.20)$$

де $Q_{p.w} = 1150 \cdot 10^3 \text{ Дж/ кг}$ – тепловий ефект реакцій піrolізу деревини; m_w – маса деревини, що вступила в реакцію піrolізу, кг; V_w - об'єм зони піrolізу, м^3 ; τ_w - тривалість процесу піrolізу, с.

Масу деревини, що вступила в реакцію піrolізу, можна визначити знаючи швидкість реакції

$$m_w = k e^{-\frac{E}{RT}} m_{all.w} \tau_w; \quad (2.21)$$

де $k = 36300$ – передекспоненційний множник, с^{-1} ; $E = (55-110) \cdot 10^3 \text{ Дж/ моль}$ – енергія активації; T – середньомасова температура зони піrolізу, К; $m_{all.w}$ - маса деревини в зоні піrolізу, кг.

Густина внутрішнього джерела теплоти, що враховує масо перенос водяної пари і її конденсацію, визначається залежністю

$$q_{V,devapor} = \frac{m_{vapor} L_{fv} k_{evac.p}}{V_{cond} \tau_{cond}} \quad (2.22)$$

де m_{vapor} - маса вологи, що випарувалася, кг; $k_{evac.p}$ - коефіцієнт, що враховує евакуацію пари в оточуюче середовище; V_{cond} - об'єм зони конденсації пари, м³; τ_{cond} - тривалість процесу конденсації, с.

У відповідності до інкрементної теорії пластичності непружні деформації ($\dot{\epsilon}_{ij}^{in}$) розглядаються як початкові. Тоді похідна по часу тензору напруги $\dot{\sigma}_{ij}$ виражається законом Гука, записаним через початкові напруги і повну деформацію ($\dot{\epsilon}_{ij}$)

$$\dot{\sigma}_{ij} = 2G\dot{\epsilon}_{ij} + \frac{2G\nu}{1-2\nu} \dot{\epsilon}_{kk} \delta_{ij} - \dot{\sigma}_{ij}^{in}, \quad i, j = \overline{1,3}, \quad (2.23)$$

де $\dot{\sigma}_{ij}^{in} = 2G\dot{\epsilon}_{ij}^{in} + \frac{2G\nu}{1-2\nu} \dot{\epsilon}_{kk} \delta_{ij} + \frac{1+\nu}{1-2\nu} \beta \delta_{ij} \dot{T}$, $i, j = \overline{1,3}$ - компоненти початкових напруг, Па; $\dot{\epsilon}_{ij} = 0,5(\dot{u}_{i,j} + \dot{u}_{j,i}) = \dot{\epsilon}_{ij}^{el} - \dot{\epsilon}_{ij}^{in}$, $i, j = \overline{1,3}$ - тензор повної швидкості деформації; \dot{u}_i , $i = \overline{1,3}$ - вектор швидкості переміщення, м; $\dot{\epsilon}_{ij}^{el}$ - пружна і $\dot{\epsilon}_{ij}^{in} = \dot{\epsilon}_{ij}^{pl} + \dot{\epsilon}_{ij}^T$ - непружна частини тензору повної швидкості деформації; $\dot{\epsilon}_{ij}^{pl}$; $\dot{\epsilon}_{ij}^T$ - тензори швидкості пластичної і температурної деформації, відповідно; $G = \frac{E}{2(1+\nu)}$ - пружний модуль зсуву, Па; E - модуль пружності при розтягу, Па; ν - коефіцієнт Пуассона; δ_{ij} - символ Кронекера; β - коефіцієнт лінійного температурного розширення, К⁻¹.

Врахування залежності електропровідності і теплопровідності сипких матеріалів кернавої пересипки і теплоізоляції печі Ачесона від тиску, пов'язаного з гравітаційним і температурним навантаженням, відбувається за допомогою моделі Дракера-Прагера, що застосовується для опису механічного стану сипучих матеріалів. У відповідності з моделлю Дракера-Прагера еквівалентна напруга визначається співвідношенням

$$\sigma_{ekv} = \beta I_1 + \sqrt{J_2}, \quad (2.24)$$

де $\beta = \frac{2\sin\varphi}{\sqrt{3}(3-\sin\varphi)}$ - константа сипкого матеріалу; φ - кут внутрішнього тертя, рад; $J_2 = \frac{1}{2}S_{ij}S_{ij}$ - другий інваріант тензору девіацій напруг, Па²; $S_{ij} = \sigma_{ij} - \frac{1}{3}I_1\delta_{ij}$, $i, j = \overline{1, 3}$ - тензор девіаторних напруг, Па; $I_1 = \sigma_{kk}$, Па.

Границя текучості сипучого матеріалу визначається залежністю

$$\sigma_{yield-str} = \frac{6ccos\varphi}{\sqrt{3}(3-\sin\varphi)} \quad (2.25)$$

де c – величина зчеплення між частинками сипкого матеріалу, Па.

Враховуючи (23) та (24), критерій текучості сипкого матеріалу можна записати у вигляді

$$F = \beta I_1 + \sqrt{J_2} - \sigma_{yield-str}. \quad (2.26)$$

В результаті рішення системи (2.13) з врахуванням критерію текучості (2.26) визначається тензор напруг и середня гідростатична напруга $\sigma_m = \frac{1}{3}I_1$ в сипкому матеріалі. Після чого розподіл тиску в сипкому матеріалі знаходиться як $p = \sigma_m - p_{var}(T)$, де p_{var} - тиск випареної вологи сипких матеріалів.

Початкові умови при $\tau = 0$:

$$\begin{cases} T = 0 \\ U = 0 \\ \dot{\sigma}_{ij}^{in} = 0, \quad i, j = \overline{1, 3} \end{cases} \quad (2.27)$$

Граничні умови при $\tau > 0$:

на торцях струмопідводів печі

$$J_n(\tau)|_{shunt} = \mathbf{n} \cdot (-\chi(p, T)\nabla U) \cup U|_{shunt} = U(\tau); \quad (2.28)$$

– на зовнішніх поверхнях печі, що межують з оточуючим середовищем

$$\begin{cases} \mathbf{n} \cdot (-\lambda(p, T)\nabla T) = \alpha_{eff}(T)(T - T_{env}); \\ \mathbf{n} \cdot \nabla U = 0; \\ \dot{\sigma} \cdot \mathbf{n} = 0; \\ \mathbf{u}(\chi) = 0, X \in X_0, \end{cases} \quad (2.29)$$

де $X_0(0, 0, 0)$ - початок координат печі, м;

- на осях симетрії печі

$$\begin{cases} \mathbf{n} \cdot \nabla T = 0; \\ \mathbf{n} \cdot \nabla U|_{axes \perp \mathbf{j}} = 0 \wedge U|_{axes \perp \mathbf{j}} = 0 \\ \mathbf{n} \cdot \nabla \mathbf{u} = 0, \end{cases} \quad (2.30)$$

де \mathbf{j} – вектор густини електричного струму, А/ м²;

- умови теплоелектричного контакту на ГК між елементами конструкції печі

$$\begin{cases} \{T\} = \mathbf{n} \cdot (r_\lambda \mathbf{q}); \\ \{\mathbf{n} \cdot \mathbf{q}\} = 0; \\ \{U\} = \mathbf{n} \cdot (r_e \mathbf{j}); \\ \{\mathbf{n} \cdot \mathbf{j}\} = 0; \end{cases} \quad (2.31)$$

– умови механічного контакту на ГК між стінками печі, заготовками і сипкими матеріалами теплоізоляції і керновій пересипки (2.32) і на інших ГК між елементами конструкції печі (2.33)

$$\begin{cases} \mathbf{n}^+ \dot{\mathbf{u}}^+ + \mathbf{n}^+ \dot{\mathbf{u}}^- - \delta \leq 0; \\ \{\dot{\sigma} \cdot \mathbf{n}\} \leq 0; \\ (\mathbf{n}^+ \dot{\mathbf{u}}^+ + \mathbf{n}^+ \dot{\mathbf{u}}^- - \delta) \dot{\sigma} \cdot \mathbf{n} = 0; \\ |\dot{\sigma}_\tau| \leq -f \dot{\sigma} \cdot \mathbf{n}; \end{cases} \quad (2.32)$$

$$\begin{cases} \{\dot{\mathbf{u}}\} = 0; \\ \{\dot{\sigma} \cdot \mathbf{n}\} = 0, \end{cases} \quad (2.33)$$

де δ – початковий проміжок по нормалі до поверхней, з якими контактують, м; $\dot{\sigma} \cdot \mathbf{n}$ – похідні по часу нормальної напруги на поверхні контакту, Па; $\dot{\sigma}_\tau$ – похідні по часу тангенційної складової напруг, Па; f – коефіцієнт сухого тертя; $\{\dot{\mathbf{u}}\} = \dot{\mathbf{u}}^+ - \dot{\mathbf{u}}^-$; $\{\dot{\sigma} \cdot \mathbf{n}\} = \dot{\sigma}^+ \cdot \mathbf{n}^+ - \dot{\sigma}^- \cdot \mathbf{n}^+$.

Перша нерівність (2.32) є умовою взаємного непроникнення тіл і пов'язано з силовим співвідношенням – друга нерівність (2.32), оскільки закриття проміжку викликає появу контактної тиску. Третя нерівність (2.32) відповідає нульовим контактним напругам, що викликає роз'єднання контактуючих тіл. Четверта нерівність (2.32) описує умови фракційної взаємодії тіл у формі закону Амонтона-Кулона.

Граничні умови (2.33) описують умови абсолютного механічного контакту між твердими тілами.

Математична модель (2.13) – (2.33) потребує рішення з п'яти рівнянь і при цьому дає змогу враховувати такі основні фактори, що визначають формування поля температур в керні, як масоперенос вологи і залежність від тиску і температури електропровідності пересипки печі. Також математична постановка (2.13) – (2.33) дає можливість розраховувати напружено-деформовані стани (НДС), що огорожують конструкції печі, і, таким чином, оцінювати довговічність роботи конструкції, наприклад, крихких графітових струмопідводів печі, як найбільш слабкого, в сенсі експлуатаційної надійності частини обладнання.

Основним критерієм оцінки достовірності отриманих результатів за допомогою математичної моделі є їх верифікація на даних натурних експериментів, що отримуються в промислових умовах на діючому пічному обладнанні.

Для чисельного рішення сформульованої задачі (2.13) – (2.33) використовується метод скінченних елементів (МСЕ), що отримав найбільше застосування для рішення широкого класу наукових та інженерних задач.

З аналізу даної системи рівнянь видно, що перше рівняння є нестационарним, а два інших квазістационарними. Таким чином, на розподіл поля температур в поточний момент часу здійснює вплив його розподіл в попередні моменти часу, в той час, як розподіл полів електричного потенціалу і механічних напруг формується миттєво, тобто не залежить від розподілу цих полів в попередні моменти часу. Основним джерелом нагріву печі Ачесона є джоулева теплота, що виділяється за рахунок проходження електричного струму через керн. Другим, за значимістю джерелом теплоти, є теплота хімічних реакцій конверсії водяної пари в керновій пересипці і окису вуглецю і водню в теплоізоляції.

У відповідності з розглядуваною системою рівнянь значення джоулевої теплоти, що виділяється в керні, залежить від квадрату напруженості електричного поля ($E = -\nabla U$, В/м) і електропровідності матеріалів, яка в свою чергу залежить від температури і тиску. З експериментальних даних по

двопараметричній залежності електропровідності $\chi(T, p)$ пересипочних вуглецевих матеріалів впливає, що в межах зміни робочої температури та тиску в сипких вуглецевих матеріалах печі Ачесона залежність електропровідності від зміни температури набагато більша, ніж від тиску [43].

2.4. Вибір методу чисельного розрахунку

Наступним кроком в математичному моделюванні процесу графітування у печах Ачесона є знаходження рішення отриманої системи рівнянь. В сучасній інженерії розрахунок таких систем проводять за допомогою числових методів. Найчастіше використовують метод скінченних елементів, що дозволяє розв'язувати системи диференціальних рівнянь в частинних похідних за допомогою рівностей, що апроксимують досліджувані рівності та є числово стабільними, тобто такими, що помилки у вхідних даних не акумулюються і не спричиняють беззмислових результатів. Основними перевагами даного методу є:

- Властивості матеріалів суміжних елементів можуть бути різними. Це дозволяє застосовувати метод до тіл, складених з декількох матеріалів;
- Скінченними елементами є прості області (прямі лінії, трикутники, прямокутники, піраміди, призми). Таким чином, даним методом можна апроксимувати тіла із складною формою країв;
- Розміри елементів можуть бути змінними. Це дозволяє збільшувати чи зменшувати елементи сітки;
- За допомогою МСЕ легко розглянути граничні умови з розривним поверхневим навантаженням, а також змішані граничні умови;
- Алгоритм методу скінченних елементів дозволяє створити загальні програми для розв'язку завдань різного класу;
- Завдання зводиться до розв'язку системи рівнянь алгебри великої розмірності. Проте хороша обумовленість системи розв'язних рівнянь алгебри дозволяє отримувати досить точні розв'язки для систем рівнянь розмірністю 5-10 мільйонів і більше.

Головний недолік цього методу полягає у потребах великого обсягу пам'яті ЕОМ і високої швидкості розрахунку. В наш час розвиток ЕОМ практично усунув цей недолік. Не дивлячись на складність даного методу в порівнянні з методом скінченних різниць, при реалізації числового розрахунку моделі за допомогою МСЕ складнощі не виникають, адже в наш час існує велика кількість програмного забезпечення призначеного для моделювання процесів, в якому вже реалізовано метод скінченних елементів.

Для чисельного розрахунку моделі графітування у печах Ачесона та графічного відображення отриманих результатів було обрано програмний засіб *ANSYS*.

2.5. Дослідження температурного печі Ачесона

На якість продукції, отриманої в результаті графітування, великий вплив мають як якість вихідної сировини, так і температурний режим процесу графітування. До основних характеристик температурного режиму процесу графітування відносяться:

- температура заготовки;
- час витримки керна при максимальній температурі;
- темп нагріву заготовок;
- перепад температури по заготовках.

Температура заготовок є найбільш важливою характеристикою процесу графітування. Від рівня температури залежить ступінь графітування, а відповідно і всі фізичні властивості кінцевої продукції. Як видно з конструкції печі Ачесона kern печі графітування характеризується значною нерівномірністю поля температур, тому, говорячи про температуру заготовок, мають на увазі максимальний рівень середньо об'ємної або мінімальної температури керна. Для отримання високоякісної продукції середньо об'ємна температура керна повинна бути не нижче 2800°C , а рівень мінімальної температури керна (торці заготовок) - не менше 2300°C .

Основний вплив на температурний режим процесу графітування здійснюють:

- енергетичний режим процесу;
- властивості теплоізоляційних матеріалів;
- геометрія керна.

Енергетичний режим є визначальним чинником температурного рівня процесу графітування і характеризується регламентом введення потужності в піч, UPE і значно залежить від властивостей керованого пересипання і геометрії керна.

Основна функція керованого пересипання полягає в збільшенні активного електричного опору керна і забезпеченні рівномірного підведення струму до графітованих заготовок. Маючи набагато більш високий опір, ніж графітові заготовки, керованого пересипання є тепловиділяючим елементом печі Ачесона. Протягом кампанії в пересипку виділяється від 96 до 98,5% всієї теплової енергії підведеної в піч, тому її фізичні властивості і спосіб застосування надає істотний вплив на розподіл енергії по керну, нагрів заготовок, хід процесу графітування і його кінцеві результати.

Для створення системи керування формуванням вуглецевих виробів необхідно дослідити вплив керувальних змінних на динаміку температурного поля. До керувальних змінних відносяться динаміка потужності печі.

В результаті числового розрахунку отриманої моделі в середовищі *ANSYS* з заданими значеннями керувальних змінних було отримано динаміку температурного поля печі Ачесона.

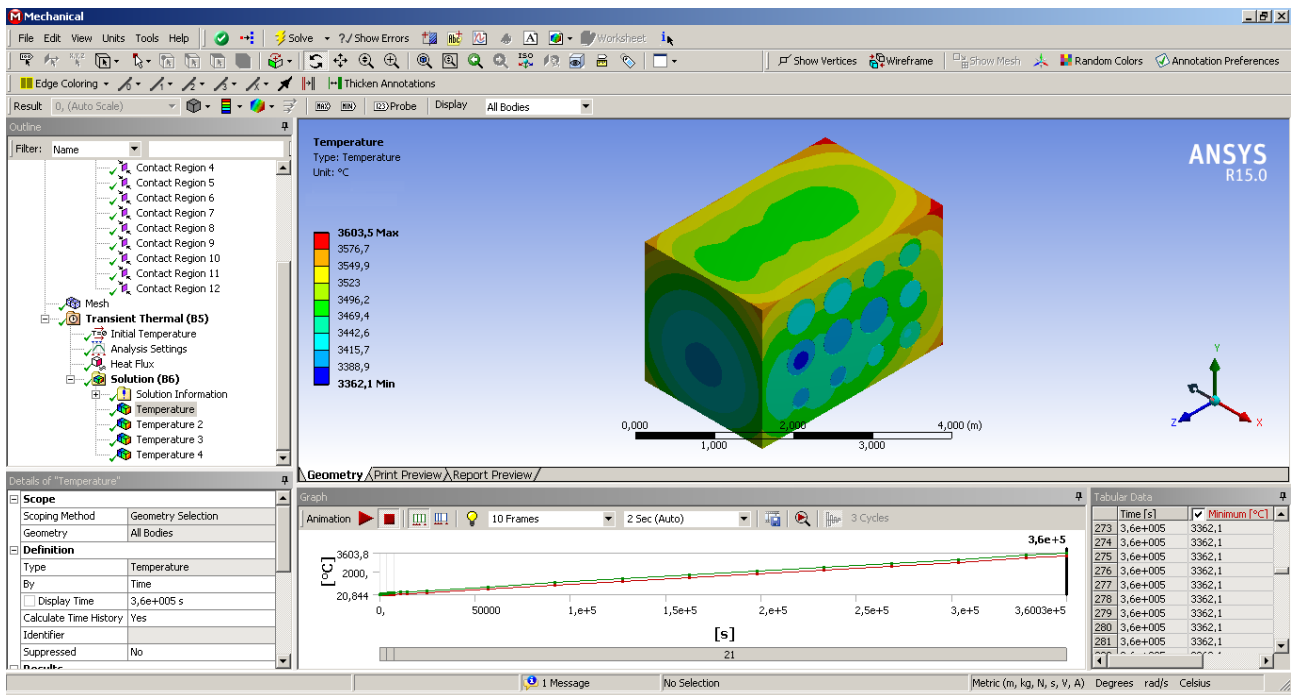


Рис.2.1. Динаміка температурних полів для заготовок різним діаметром

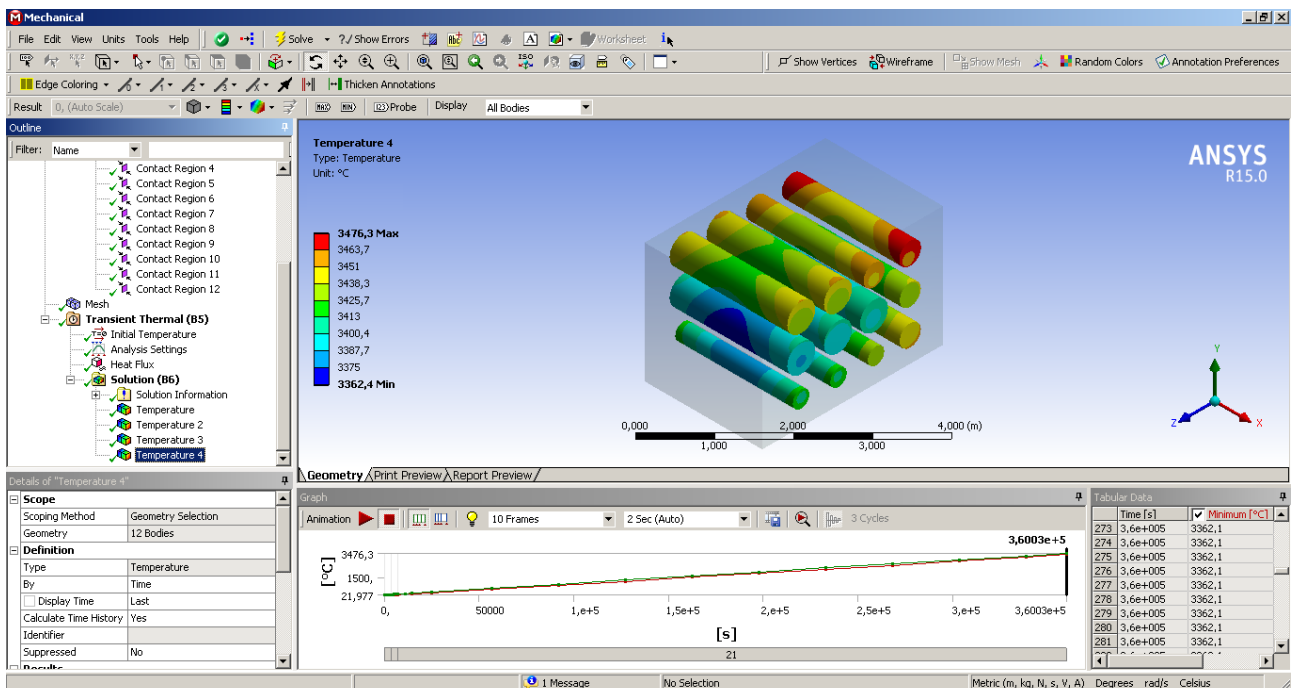


Рис.2.2. Динаміка температурних полів для заготовок різним діаметром

3. СПРОЩЕННЯ МОДЕЛІ ПРОЦЕСУ

3.1. Спрощення моделі процесу

Процеси печі графітуванні мають багато різноманітних варіантів математичних моделей, але вони не можуть використовуватися у системах керування в реальному часі, оскільки їхній розрахунок може тривати близько 3 діб. Така тривалість розрахунку не дозволяє досліджувати системи керування процесом графітування у печах Ачесона, тому було прийняте рішення розробити спрощену модель, в якій час розрахунку буде значно меншим, що дозволить симулювати та досліджувати роботу систем керування цим процесом.

Спрощені моделі процесу наразі – це моделі, які зазвичай мають досить високу точність повторення експериментальної моделі, але вони вимагають у рази менше часу на свою реалізацію. Справжнє виробництво не може надати для випробовувань і проведення розрахунків 3 доби, адже це понесе за собою великі економічні витрати. Тому, для розробок, модернізацій та моделювання краще використовувати спрощені моделі процесу.

У даному розділі реалізовано знаходження спрощеної моделі, що може оцінювати температуру в контрольних точках з малою похибкою по відношенню до складної моделі і доволі швидко розраховується.

3.2. Аналіз існуючих методів спрощення моделі

На сьогоднішній день існує безліч підходів до спрощення моделей [44], з яких найчастіше використовується метод лінеаризації нелінійних диференціальних рівнянь, методи ідентифікації, метод Фур'є.

Метод лінеаризації передбачає зведення нелінійних диференціальних рівнянь до їх лінійного наближення. Розв'язок системи лінійних диференціальних рівнянь - задача простіша ніж розв'язок системи нелінійних диференціальних рівнянь і розрахунок моделі представленої в такому вигляді займає набагато менше часу. Недоліками цього методу є те, що далеко не всі нелінійні диференціальні рівняння піддаються лінеаризації та процеси, що мають явно

виражений нелінійний характер будуть досить неточно описуватись такими моделями.

Методи ідентифікації дозволяють за отриманими зі складної моделі даними розрахувати коефіцієнти спрощеної моделі. Складнощі в застосуванні таких методів полягають в необхідності підбирати структуру моделі. Недолік таких методів полягає в тому, що навіть при підбраній оптимальній структурі моделі зі збільшенням числа вихідних сигналів точність моделі значно зменшується.

Метод Фур'є передбачає представлення вихідних сигналів у вигляді ряду, що складається з суми добутків коефіцієнтів, що відображають зміну вихідних сигналів в часі та коефіцієнтів, що відображають розподілення вихідних сигналів в просторі. В таких моделях залежність вихідних сигналів від вхідних визначається ідентифікацією залежності значень коефіцієнтів, що відображають зміну вихідних сигналів в часі від вхідних сигналів. Моделі, отримані за методом Фур'є не мають недоліку моделей, створених за допомогою методів ідентифікації, адже кількість членів ряду, що необхідні для точного відображення вихідних сигналів (а отже і кількість коефіцієнтів, що відображають зміну вихідних сигналів в часі) менша за кількість вихідних сигналів.

Розглянувши дані методи було прийнято рішення використовувати для спрощення моделі метод Фур'є, адже він найбільш універсальний з розглянутих методів.

3.3. Розроблення спрощеної моделі за допомогою методу Фур'є

3.3.1. Метод Фур'є

Спрощення моделі полягає у апроксимації ANSYS-моделі простішими моделями (в цьому випадку, моделями з меншою кількістю рівнянь), використовуючи залежності між температурами в різних точках сітки. Метою цього є знаходження спрощеної моделі, що може передбачувати/оцінювати температури з малою похибкою по відношенню до ANSYS-моделі.

Спрощена модель отримується з ряду, так званих, «знімків», що відображають температурні профілі в кожен момент часу. Таким чином температура $T(\xi, t)$ подається у вигляді ряду ортонормованих базисних функцій $\varphi_i(\xi)$ (під ортонормованістю $\{\varphi_{T,i}(\xi)\}_{i=1}^{\infty}$ вважається, що $\langle \varphi_{T,i}, \varphi_{T,j} \rangle = 0$ і $\langle \varphi_{T,i}, \varphi_{T,i} \rangle = 1$)

$$T(\xi, t) = \sum_{i=1}^{\infty} a_i(t) \varphi_i(\xi) \quad (3.1)$$

Кожна з яких помножена на функцію від часу

$$a_i(t) = \langle T(\cdot, t), \varphi_i(\cdot) \rangle \quad (3.2)$$

яка називається коефіцієнтом Фур'є. Такий ряд називається розкладом Фур'є. Скінченний ряд довжиною n ,

$$\tilde{T}(\xi, t) = \sum_{i=1}^n a_i(t) \varphi_i(\xi) \quad (3.3)$$

застосовується для апроксимації температури $T(\xi, t)$. Базисні вектори $\varphi_i(\xi)$ отримуються із даних ANSYS-моделі. Після цього, рівняння моделі мають бути перетворені на залежність між входами моделі $u(t)$ і коефіцієнтами Фур'є $\{a_i(t)\}_{i=1}^n$. На цьому етапі рівняння моделі отримуються завдяки алгоритмам ідентифікації. Процес спрощення моделі поділяється на два етапи:

1. Визначення ряду базисних векторів $\{\varphi_i(\xi)\}_{i=1}^n$ на основі набору даних симуляції $\{\tilde{u}(t_k), \tilde{T}(\xi, t_k)\}_{k=1}^K$, де K – кількість кроків по часу у симуляції. Знак \sim означає, що дані отримані із симуляції.

2. Вибір структури моделі залежності між $u(t)$ та $\{a_i(t)\}_{i=1}^n$ та визначення невідомих параметрів цієї моделі на основі даних $\{\tilde{u}(t_k)\}_{k=1}^K$ та $\{\tilde{a}_1(t_k), \dots, \tilde{a}_n(t_k)\}_{k=1}^K$, де $\tilde{a}_i(t_k)$ можуть бути розраховані із $\varphi_i(\xi)$ та $\tilde{T}(\xi, t_k)$. Після того, як будуть визначені всі параметри, буде отримана спрощена модель, за допомогою якої можна передбачати зміну в часі коефіцієнтів Фур'є із заданої траєкторії вхідного сигналу $u(k)$.

3.3.2. Визначення базисних векторів та коефіцієнтів Фур'є

Далі в роботі буде використовуватись наступний запис:

$$a(k) = \text{col}\{a_i(t_k)\}_{i=1}^N \quad (3.4)$$

$$T(k) = \text{col}\{\tilde{T}(\xi_l, t_k)\}_{l=1}^N \quad (3.5)$$

$$\varphi_i = \text{col}\{\tilde{\varphi}_i(\xi_l)\}_{i=1}^N \quad \Phi = (\varphi_1 \quad \varphi_2 \quad \dots \quad \varphi_N) \quad (3.6)$$

Із врахуванням цього, рівняння (3.3) можна записати так:

$$T(k) = \Phi a(k) \quad (3.7)$$

Оскільки стовпчики Φ формують ортонормований базис, то матриця Φ є ортогональною, що означає $\Phi^T \Phi = I_N$, де I_N – одинична матриця $N \times N$. Права частина рівняння 3.7 містить N базисних векторів і N коефіцієнтів Фур'є. Цей ряд базисних векторів $\{\varphi_i\}_{i=1}^N$ та відповідний ряд коефіцієнтів Фур'є $\{a_i(k)\}_{i=1}^N$ розділені на ряд із n базисних векторів $\{\varphi_i\}_{i=1}^n$ із коефіцієнтами, які визначають апроксимацію $\tilde{T}(k) := \sum_{i=1}^n a_i(k) \varphi_i$ та ряд із $N - n$ базисних векторів $\{\varphi_i\}_{i=n+1}^N$ із коефіцієнтами, що формують ряд Фур'є вектора похибок $\varepsilon_n(k) := T(k) - \tilde{T}(k) = \sum_{i=n+1}^N a_i(k) \varphi_i$.

Далі використовується ℓ_2 – норма $\|\cdot\|_2$, що обраховується за формулою:

$$\|x\|_2 = (\sum_{k=1}^K x^T(k)x(k))^{1/2} \quad (3.8)$$

Через ортогональність матриці Φ , вага кожного елемента $\varphi_i a_i$ визначається коефіцієнтами:

$$\begin{aligned} \|T\|_2^2 &= \sum_{k=1}^K T^T(k)T(k) = \sum_{k=1}^K a^T(k)\Phi^T\Phi a(k) = \|a\|_2^2 \\ &= \|a_1\|_2^2 + \|a_2\|_2^2 + \dots + \|a_N\|_2^2 \end{aligned}$$

Для заданих температур $\{\tilde{T}(k)\}_{k=1}^K$ необхідно підібрати такий базис $\{\varphi_i\}_{i=1}^N$, ℓ_2 – норма коефіцієнтів Фур'є якого задовольняє

$$\|a_1\|_2^2 \geq \|a_2\|_2^2 \geq \dots \geq \|a_N\|_2^2 \quad (3.9)$$

Це означає, що для будь-якої довжини ряду n , «енергія» $\sum_{i=1}^n \|a_i\|$ в перших n коефіцієнтах Фур'є є максимальною.

Таким чином, якщо ряд Фур'є обмежується n – членами так, що:

$$T(k) = \Phi a(k) = [\Phi_n \quad \Phi_{\text{tail}}] \begin{bmatrix} a_n(k) \\ a_{\text{tail}}(k) \end{bmatrix}$$

, то квадрат ℓ_2 – норми вектора температур можна записати як

$$\|T\|_2^2 = \|a_n\|_2^2 + \|a_{\text{tail}}\|_2^2 = \|a_n\|_2^2 + \|\varepsilon_n\|_2^2$$

де $a_n(k) \in \mathbb{R}^n$ – вектор, що містить перші n коефіцієнтів Фур'є, $a_{tail}(k) \in \mathbb{R}^{N-n}$ – вектор, що містить останні $N - n$ коефіцієнтів Фур'є та $\varepsilon_n(k) \in \mathbb{R}^N$, $\varepsilon_n(k) = \phi_{tail} a_{tail}$ – вектор похибок, отриманих внаслідок обмеження ряду.

Набір базисних векторів, які дають малу ℓ_2 – норму $\varepsilon_n(k)$ визначається із ряду «знімків» $\{\tilde{T}(k)\}_{k=1}^K$, отриманих із ANSYS-моделі при впливі ряду вхідних даних $\tilde{y}(k)$. Потрібно зазначити, що для заданого набору даних $\{\tilde{T}(k)\}_{k=1}^K$ і для будь-якого n , базис мінімізує ℓ_2 – норму відповідного вектора похибок $\tilde{\varepsilon}_n(k) = \tilde{T}(k) - \check{T}(k)$ із усіх ортонормованих базисів n -ного порядку. Спочатку стверджуємо, що ℓ_2 – норму вектора похибок $\varepsilon_n(k)$ можна мінімізувати шляхом максимізації обмеженого вектора коефіцієнтів Фур'є $a_n(k)$. Таким чином, ℓ_2 – норма вектора похибок обмеження $\tilde{\varepsilon}_n(k)$ мінімізується шляхом вибору таких базисних векторів $\{\varphi_i\}_{i=1}^n$, при яких максимізується:

$$F(\varphi_1, \varphi_2, \dots, \varphi_n) = \sum_{i=1}^n \|\tilde{T}^T(k) \varphi_i\|_2^2 \quad (3.10)$$

Якщо

$$\|\varphi_i\|_2^2 = 1 \quad (3.11)$$

та

$$\langle \varphi_i, \varphi_j \rangle = 0 \text{ при } i \neq j \quad (3.12)$$

, де $\|\varphi_i\| := \varphi_i^T \varphi_i$ – ℓ_2 – норма базисного вектора φ_i . Оптимальні базисні вектори $\{\varphi_i\}_{i=1}^n$ мають бути власними векторами матриці $R := T_{snap} T_{snap}^T$. Також, базисні вектори задовольняють умови ортонормованості у випадку, якщо власні значення з відповідних власних векторів є різними. Для виведення оптимуму $F(\varphi_1, \varphi_2, \dots, \varphi_n)$ при $\|\varphi_i\|^2 = 1$, запишемо Лагранжан

$$\mathcal{L}(\varphi_1, \dots, \varphi_n, \lambda_1, \dots, \lambda_n) = F(\varphi_1, \dots, \varphi_n) - \sum_{i=1}^n \lambda_i (\|\varphi_i\|^2 - 1) \quad (3.13)$$

Базисні вектори $\varphi_1, \varphi_2, \dots, \varphi_n$ потрібно обрати такими, щоб максимізувати \mathcal{L} . Необхідною умовою екстремуму для цієї функції є те, щоб градієнти Лагранжана по базисним векторам φ_i та множникам Лагранжа λ_i дорівнювали нуль-вектору

$$\nabla_{\varphi_i} \mathcal{L} = 0^T \quad \nabla_{\lambda_i} \mathcal{L} = 0 \quad i = 1, 2, \dots, n \quad (3.14)$$

де $\nabla_{\varphi_i} = \frac{\partial}{\partial \varphi_i^T}$ – градієнт за i -м базисним вектором, 0^T – нульовий вектор-рядок такого ж розміру, як $\nabla_{\varphi_i} \mathcal{L}$. Із врахуванням формули для ℓ_2 – норми, Лагранжан можна записати у вигляді:

$$\mathcal{L}(\varphi_1, \dots, \varphi_n, \lambda_1, \dots, \lambda_n) = \sum_{i=1}^n \sum_{k=1}^K \varphi_i^T \tilde{T}(k) \tilde{T}^T(k) \varphi_i - \sum_{i=1}^n \lambda_i (\varphi_i^T \varphi_i - 1) \quad (3.15)$$

Зазначивши, що $\tilde{T}(k)$ – k -та колонка матриці $T_{snap}(k)$ отримаємо

$$\mathcal{L}(\varphi_1, \dots, \varphi_n, \lambda_1, \dots, \lambda_n) = \sum_{i=1}^n \varphi_i^T T_{snap}(k) T_{snap}^T(k) \varphi_i - \sum_{i=1}^n \lambda_i (\varphi_i^T \varphi_i - 1) \quad (3.16)$$

Градієнти Лагранжана:

$$\nabla_{\varphi_i} \mathcal{L} = \varphi_i^T R - \lambda_i \varphi_i^T = 0^T \quad (3.17)$$

$$\nabla_{\lambda_i} \mathcal{L} = \varphi_i^T \varphi_i - 1 = 0 \quad (3.18)$$

Матриця $R = T_{snap}(k) T_{snap}^T(k)$ називається матрицею кореляції. Із умов оптимальності випливає, що λ_i є власними значеннями матриці R , тоді

$$R \varphi_i = \lambda_i \varphi_i \quad (3.19)$$

Домноживши (3.18) на φ_i^T , отримуємо

$$\varphi_i^T \lambda_i \varphi_i = \varphi_i^T R \varphi_i = \varphi_i^T T_{snap} T_{snap}^T \varphi_i \quad (3.20)$$

$$\lambda_i = \|T_{snap}^T \varphi_i\|^2 = \sum_{k=1}^K |\tilde{T}^T(k) \varphi_i|^2 = \|\tilde{T}^T \varphi_i\|_2^2 = \|a_i\|_2^2 \quad (3.21)$$

Розташуємо власні значення у порядку:

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N \quad (3.22)$$

Після чого, оптимальне значення критерію визначається сумою власних значень λ_i ,

$$\|a_n^*\|_2^2 = \sum_{i=1}^n \sum_{k=1}^K |a_i^*(k)|^2 = \sum_{i=1}^n \|a_i^*\|_2^2 = \sum_{i=1}^n \lambda_i$$

де a_n^* – вектор оцінок сигналу a_n , що максимізує Лагранжан \mathcal{L} . Крім того, власні вектори $\varphi_1, \varphi_2, \dots, \varphi_n$ задовольняють умовам ортонормованості, якщо власні значення – різні. Оптимальний набір базисних векторів можна знайти, виконавши декомпозицію власних значень,

$$\begin{aligned} [R \varphi_1 \quad R \varphi_2 \quad \dots \quad R \varphi_N] &= [\lambda_1 \varphi_1 \quad \lambda_2 \varphi_2 \quad \dots \quad \lambda_N \varphi_N] \\ R \Phi &= \Phi \Lambda_R \\ R &= \Phi \Lambda_R \Phi^T \end{aligned} \quad (3.23)$$

де Λ_R – діагональна матриця власних значень λ_i матриці R (позначається “diag”), Φ – ортогональна матриця власних векторів матриці R .

Прямий розрахунок декомпозиції власних значень для прикладних задач, на ПК неможливий через великий розмір матриці R . Для більш раціонального знаходження базисних векторів можна використовувати сингулярну декомпозицію (SVD) матриці T_{snap} .

Матриця знімків T_{snap} , що містить спостережені значення $\tilde{T}(k)$, підлягає розкладу за сингулярними значеннями,

$$T_{snap} = \Phi_K \Sigma \Psi^T = \sum_{i=1}^{rank(T_{snap})} \sigma_i \varphi_i \psi_i^T \quad (3.24)$$

де $\Phi_K \in \mathbb{R}^{N \times K}$ та $\Psi \in \mathbb{R}^{K \times K}$ – ортонормовані матриці, такі, що $\Phi_K^T \Phi_K = \Psi^T \Psi = I_K$. Де I_K – одинична матриця розміром $K \times K$. Вектори φ_i та ψ_i – стовпчики матриць Φ_K та Ψ , відповідно. Такий розклад зазвичай застосовується в системному аналізі. Матриця Σ є діагональною матрицею, елементи якої, σ_i , називаються сингулярними числами матриці T_{snap} . Важливим параметром SVD є залежність між SVD розкладом і двома розкладами за власними значеннями,

$$R = T_{snap} T_{snap}^T = \Phi_K \Sigma \Psi^T \Psi \Sigma \Phi_K^T = \Phi_K \Sigma^2 \Phi_K^T \quad (3.25)$$

$$R' = T_{snap}^T T_{snap} = \Psi \Sigma \Phi_K^T \Phi_K \Sigma \Psi^T = \Psi \Sigma^2 \Psi^T \quad (3.26)$$

де $\Sigma^2 = \Sigma \Sigma$ – діагональна матриця, що складається з перших K – власних значень матриці R , стовпці Φ_K є першими K – власними векторами матриці R , стовпці Ψ є першими K – власними векторами матриці $R' \in \mathbb{R}^{K \times K}$. Перші K – власних значень матриці R можуть бути розраховані в результаті розкладу за власними значеннями матриці R' :

$$R' = \Psi \Lambda_{R'} \Psi^T$$

Обрахувавши перші K власних значень та власних векторів матриці R' базисні вектори можна вираховувати за формулою:

$$\varphi_i = T_{snap} \psi_i (\lambda_i)^{-1/2}$$

, де ψ_i – є i -им стовпчиком матриці Ψ

Отже алгоритм знаходження базисних векторів включає такі пункти:

- За отриманими раніше температурними знімками T_{snap} визначити матрицю $R' = T_{snap}^T T_{snap}$;
- Визначити власні значення $\Lambda_{R'}$ та власні вектори Ψ матриці R' ;
- Обчислити значення базисних векторів $\varphi_i = T_{snap} \psi_i(\lambda_i)^{-1/2}$.

Після визначення базисних векторів обчислюються коефіцієнти Фур'є

$$A_{snap} = [\tilde{a}_n(1) \quad \tilde{a}_n(2) \quad \dots \quad \tilde{a}_n(K)] = \Phi_K^T T_{snap} \quad (3.27)$$

В результаті розкладу за сингулярними числами були отримані значення базис векторів φ_i (табл. 3.1.).

Таблиця 3.1. Значення базис-векторів, що використовуються для апроксимації

$\xi \backslash i$	1	2	3	4	5	6	7	8	9
1	-0,44227	-0,84041	-0,04605	-0,21071	0,03332	0,17211	0,13821	0,02830	-0,030477
2	-0,50177	0,20548	-0,68476	0,27246	-0,03679	0,03717	-0,27245	0,29167	-0,028699
3	-0,59658	0,41497	0,25761	-0,54178	0,26431	-0,15551	0,04075	-0,11021	0,06427
4	-0,03238	-0,10695	-0,01986	-0,18520	-0,45452	-0,11651	-0,67550	-0,52515	0,02354
5	-0,03387	-0,05391	-0,07592	-0,05602	-0,44798	-0,50134	0,28778	0,19337	0,64403
6	-0,03319	-0,01384	0,01519	-0,09599	-0,32560	-0,50122	0,16701	0,18246	-0,75553
7	-0,15817	-0,20829	0,34795	0,50548	0,40884	-0,52653	-0,32328	0,04247	0,06544
8	-0,34310	0,11464	0,08465	0,51038	-0,22912	0,09904	0,44347	-0,58218	-0,05970
9	-0,22507	0,09056	0,57270	0,15867	-0,44126	0,37630	-0,19183	0,46342	0,00962

3.3.3 Ідентифікація залежностей між отриманими значеннями коефіцієнтів Фур'є та вхідними сигналами

Для ідентифікації залежностей будемо використовувати пакет *System Identification Toolbox*, що входить до складу *Matlab*. Пакет *System Identification Toolbox* містить засоби для створення математичних моделей лінійних динамічних об'єктів (систем) на основі спостережуваних вхідних/вихідних даних. Для цього він має зручний графічний інтерфейс, що допомагає

організувати дані й створювати моделі без використання функцій командного режиму роботи. Точніше кажучи, ці функції використовуються, насамперед, для підтримки моделей у середовищі графічного інтерфейсу *GUI*. Методи ідентифікації, що входять у пакет, застосовні для рішення широкого класу завдань – від проектування систем керування й обробки сигналів до аналізу часових рядів.

Основні властивості пакета наступні:

1. простий і гнучкий інтерфейс;
2. попередня обробка даних, включаючи фільтрацію, видалення трендів і зсувів;
3. вибір діапазону даних для аналізу;
4. ефективні методи авторегресії;
5. можливості аналізу відгуку систем у часовій і частотній областях;
6. відображення нулів і полюсів передаточної функції системи;
7. аналіз нев'язань при тестуванні моделі.

В результаті ідентифікації для кожного з $n=1..9$ було знайдено такі моделі, що найточніше описують зв'язок між значеннями коефіцієнтів $a_i(t)$ та вхідними сигналами моделі. У всіх випадках найкраще з поставленою задачею впоралися ARX моделі авторегресії зі структурою типу:

$$y(t) + a_1 y(t-1) + \dots + a_{na} y(t-na) = b_1 u(t-nk) + \dots + b_{nb} u(t-nb-nk+1) + e(t)$$

В таблиці 3.2 містяться дані про конфігурацію оптимальних ідентифікаційних моделей для кожного з n та їх точність.

Таблиця 3.2. Дані про оптимальні ідентифікаційні моделі

n	na	nb	nk	Точність моделі, %
1	3	2	1	88,40
2	5	3	1	65,70
3	1	4	5	85,75
4	1	5	1	76,02
5	5	2	4	69,62

6	5	2	4	67,07
7	5	2	4	53,69
8	5	2	4	49,80
9	5	3	3	53,64

Як видно з вищенаведених даних точність ідентифікаційних моделей зменшується зі збільшенням n , адже збільшується кількість виходів моделі і характер зв'язків між входами і виходами стає складнішим.

3.3.4 Дослідження якості роботи спрощених моделей

Для отримання спрощених моделей матрицю, що складається з n базис векторів помножуємо на вектори вихідних сигналів ідентифікаційних моделей (значення коефіцієнтів $\alpha_i(t)$ в момент часу t).

Сума середньоквадратичних похибок утворень вихідних параметрів для кожної моделі представлені в таблиці 3.3.

Таблиця 3.3. Середньоквадратичні похибки утворень вихідних параметрів

n	1	2	3	4	5	6	7	8	9
Похибка	49,270	39,818	21,145	17,950	17,160	15,022	15,671	14,121	14,215

Проаналізувавши таблицю 3.3. – можна побачити, що похибка на 8 ітерації – найменша, тому для зменшення «захаращеності» графіків проаналізуємо відхилення від даних нам значень експериментальні дані з 8 ітерації.

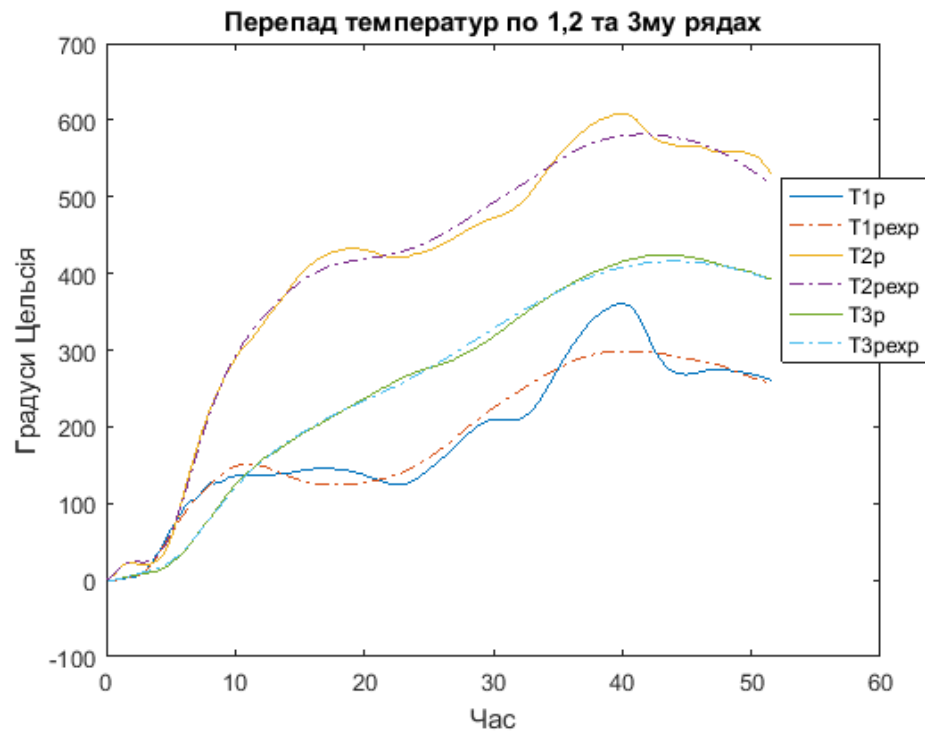


Рис.3.3. Значення перепадів температур по 3 рядах, що отримані з складної моделі та з спрощених моделей

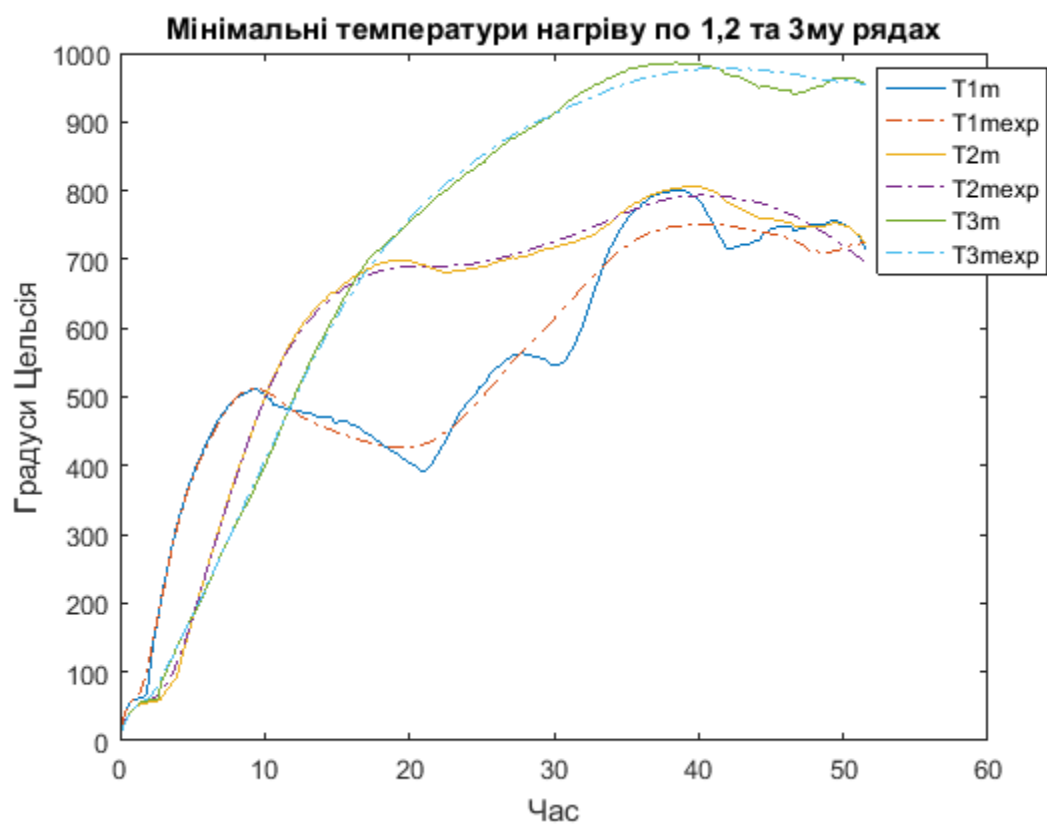


Рис.3.4. Значення мінімальних температур по 3 рядах, що отримані з складної моделі та з спрощених моделей

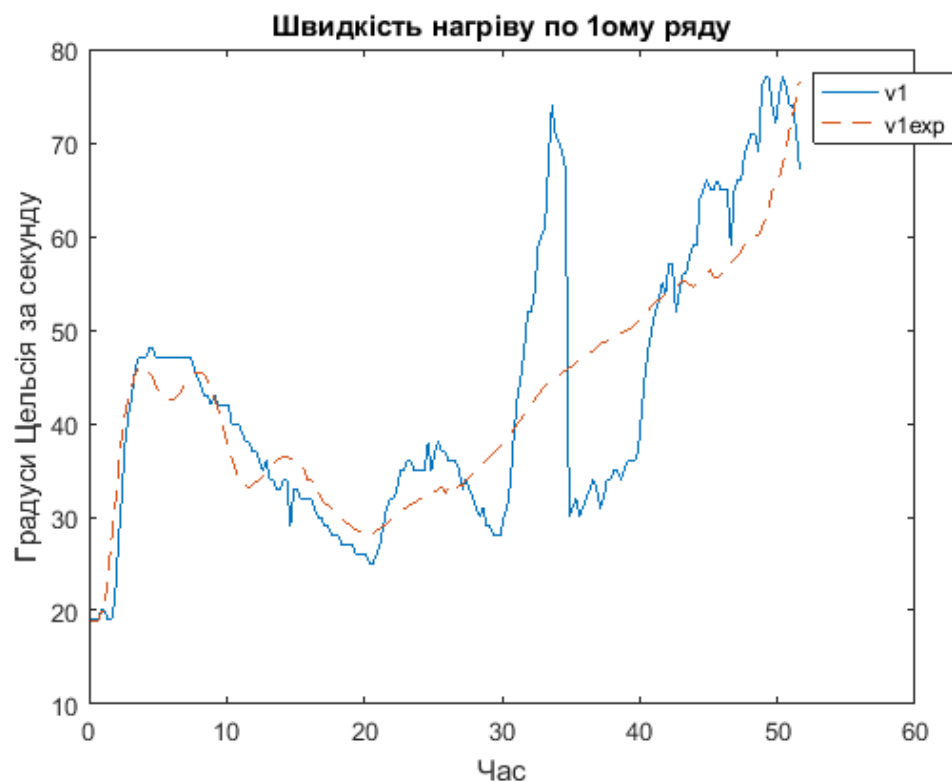


Рис.3.5. Значення швидкості нагріву по 1 ряду, що отримані з складної моделі та з спрощених моделей

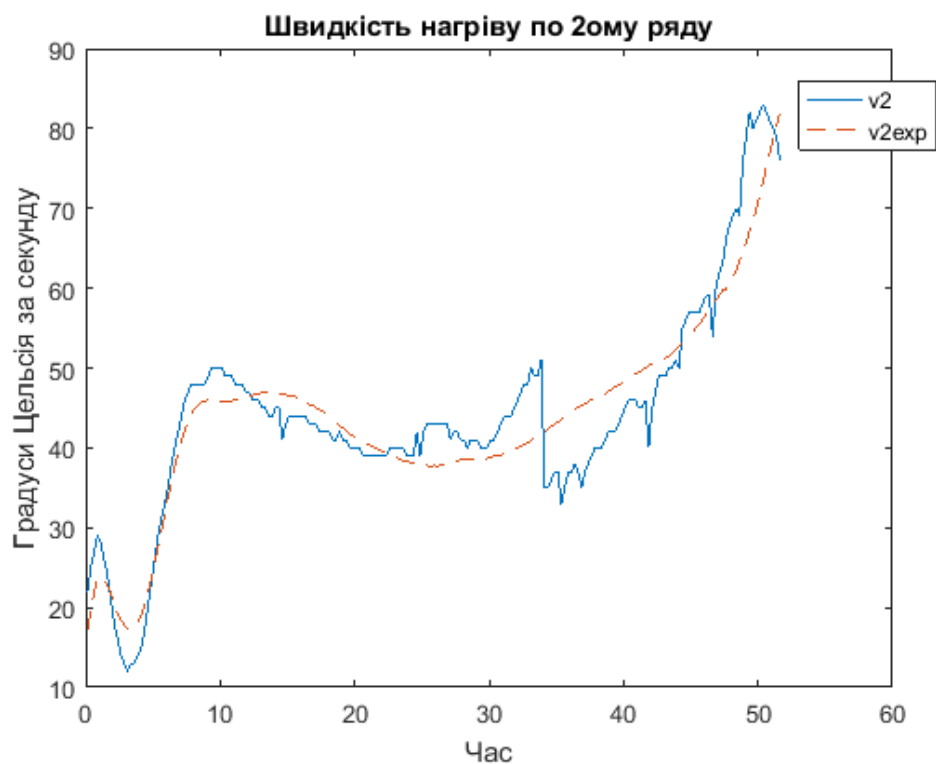


Рис.3.6. Значення швидкості нагріву по 2 ряду, що отримані з складної моделі та з спрощених моделей

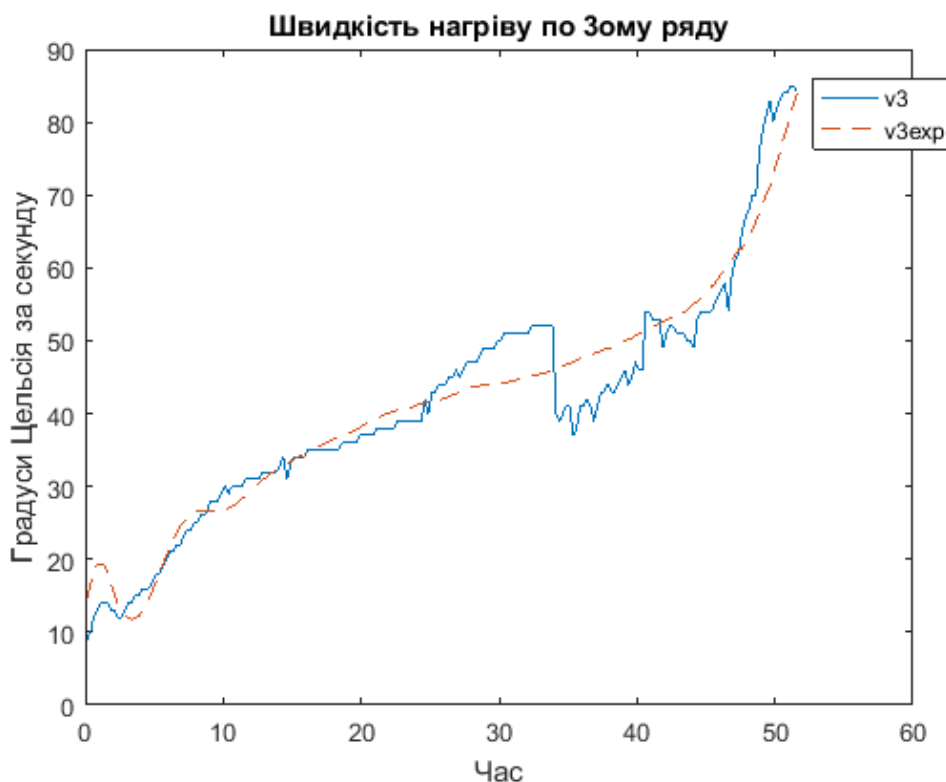


Рис.3.7. Значення швидкості нагріву по 3 ряду, що отримані з складної моделі та з спрощених моделей

Як вже було зрозуміло зі значення похибок, найкраще з усіх моделей працює модель з $n=8$. Це спричинено тим, що попри високу точність апроксимації модель з $n=9$ має досить велику похибку при ідентифікації порівняно з іншими моделями.

Отже в даному розділі отримано спрощену модель процесу печі графітування, що доволі точно відтворює температуру в контрольних точках і буде використана в дослідженні систем керування цим процесом в наступному розділі.

4. СИНТЕЗ СИСТЕМИ КЕРУВАННЯ ПІЧЧЮ ГРАФІТУВАННЯ

4.1. Керування процесом графітуванням у печах Ачесона

Основною задачею розділу є створення оптимальної системи керування печею Ачесона, яка б забезпечувала оптимальний вихід бездефектної продукції. Для цього необхідно, щоб під час процесу графітування були дотримані температурні умови.

Автоматизована система управління розробляється для автоматичного керування технологічним процесом графітування електродів.

Система складається з автоматизованої системи керування технологічним режимом графітування: автоматичне керування силовими агрегатами (включення, відключення, перемикання ступенів і т.п.), ведення технологічного режиму графітування згідно заданим режимам, контроль і фіксація технологічних параметрів згідно з технологічною інструкцією затвердженою на заводі.

4.2. Способи керування графітуванням у печах Ачесона

Спосіб контролю теплового режиму процесу графітування. Шляхом вимірювання температури в початковій стадії розігріву печі з вимірюванням електричного опору і обчисленням температури на наступних стадіях графітування як величиною, обернено пропорційною електричному опору печі. Недоліком цього способу є те, що, перш за все, через індивідуальні особливості вуглецевих матеріалів електричний опір печі не завжди характеризує досягнуту температуру. Причому на завершальній стадії процесу характер зміни електричного опору настільки незначний, що не може служити критерієм для управління режимом роботи печі. Крім того, розрахунок температури проводиться для конкретної певної точки простору печі, що не дає інформації про процес утворення тріщин в заготівлі в процесі роботи печі. Все це не дозволяє точно формувати величину струму, що підводиться до печі і, відповідно, забезпечити стабілізацію якості електродів на стадії графітування.

В даний момент управління процесом графітування в печах Ачесона, в основному виконується шляхом завдання регламенту введення потужності і питомої витрати електроенергії. Такий спосіб найбільш простий в реалізації і не вимагає унікального приладового забезпечення, та дозволяє отримати точне уявлення про головне критерії процесу графітування - температури, її розподілу і швидкості зміни [45].

4.3. Проектування системи керування піччю графітування

Аналіз існуючих систем керування піччю графітування показує, що під час кампанії графітування відбувається значне акумулювання теплоти в пересипці, та можливе графітування за рахунок цієї теплоти.

Так як кампанія графітування відзначається своєю енергозатратністю, тому навіть не значне зменшення затрат призводить до значного прибутку, тому в даному розділі пропонується провести дослідження можливості реалізації наведеного вище твердження та розробки системи керування з передчасним вимкненням живлення, при якому забезпечується отримання виробів відповідної якості.

Розробка системи керування графітування базується на розроблену у роботі [3] критерію, а саме ступеню графітування, який дозволяє стверджувати, що ступінь графітування буде найменшим в “найхолоднішому” місці виробів.

$$\beta = \int_{\tau_1}^{\tau_2} \frac{d\tau}{f(T(\tau))}, \quad (4.1)$$

де τ_1, τ_2 - часовий інтервал в якому температура характеристичної точки виробів $T(\tau)$ перевищує 2000 °C, f - температурна залежність тривалості графітування.

На рис. 4.1 зображено залежність часу графітування від температури.

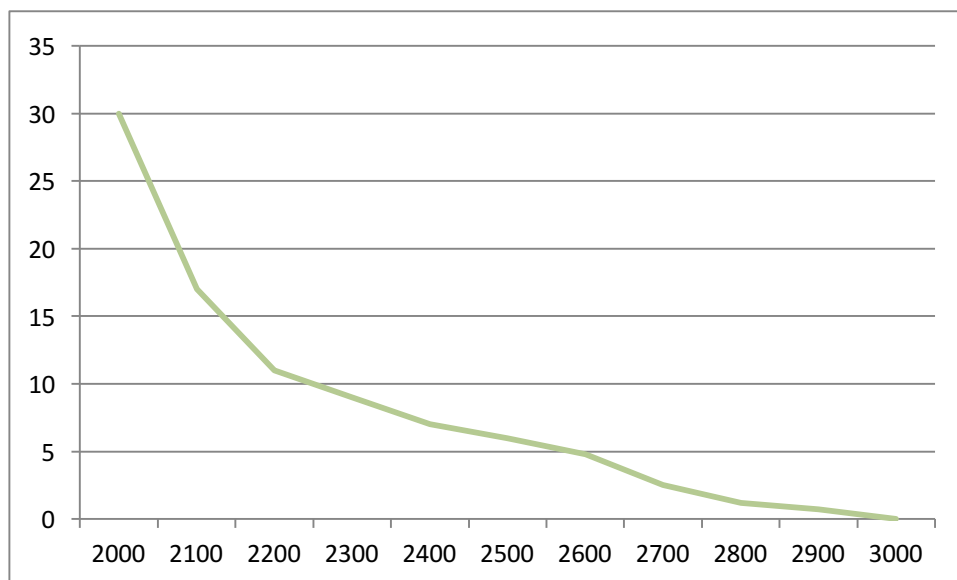


Рис. 4.1 - залежність часу графітування від температури

Так як засипка акумулює в собі значну кількість теплоти, можливо говорити про те, що заготовка продовжує графітуватись навіть після припинення підводу теплоти.

Саме тому, існує можливість розробки системи керування, що буде забезпечувати розрахунок поточного ступеня графітування та на його основі приймати рішення щодо доцільності відключення живлення.

Для розробки описаної вище системи керування необхідно провести досліди, в результаті яких буде отримано залежність «дографітування» в залежності від поточної температури.

Для цього необхідно вивести температуру на усталений рівень, наприклад 2300 °C та визначити час, при якому ступінь графітування буде $\beta = 0,98$, тобто заготовка буде прографітована.

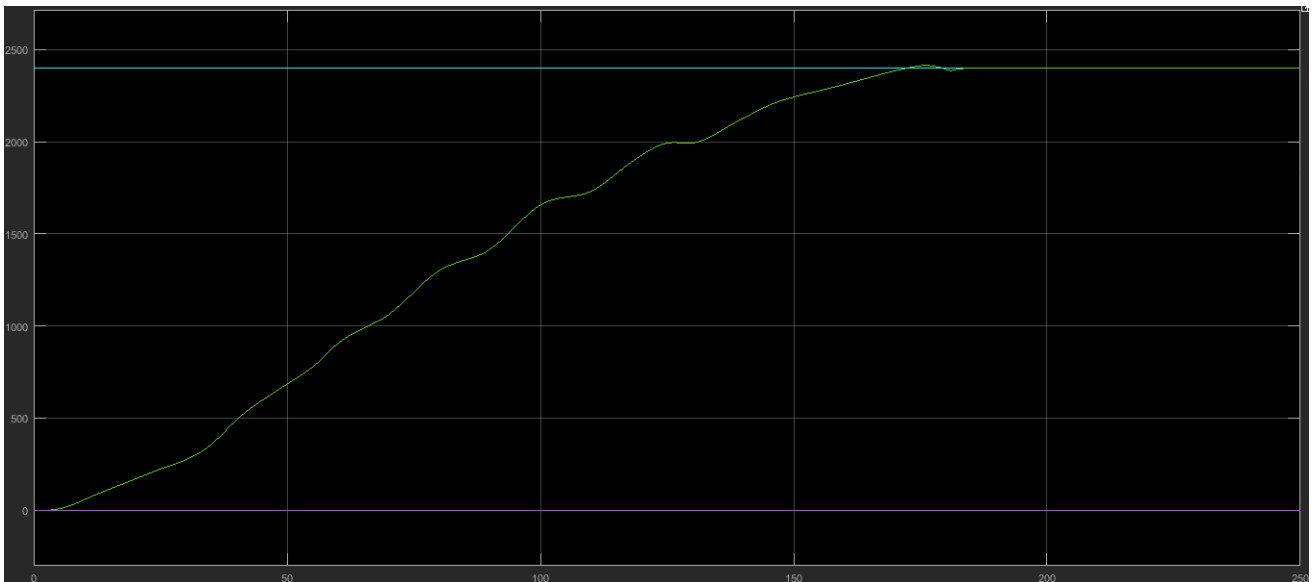


Рис. 4.2 - перехідна характеристика зміни температури по каналу керування протягом за 250 ітерацій (одна ітерація – 0,25 годин)

Для розрахунку ступеня графітування було отримано реалізовано схему, що реалізує розрахунок ступеня графітування [4]:

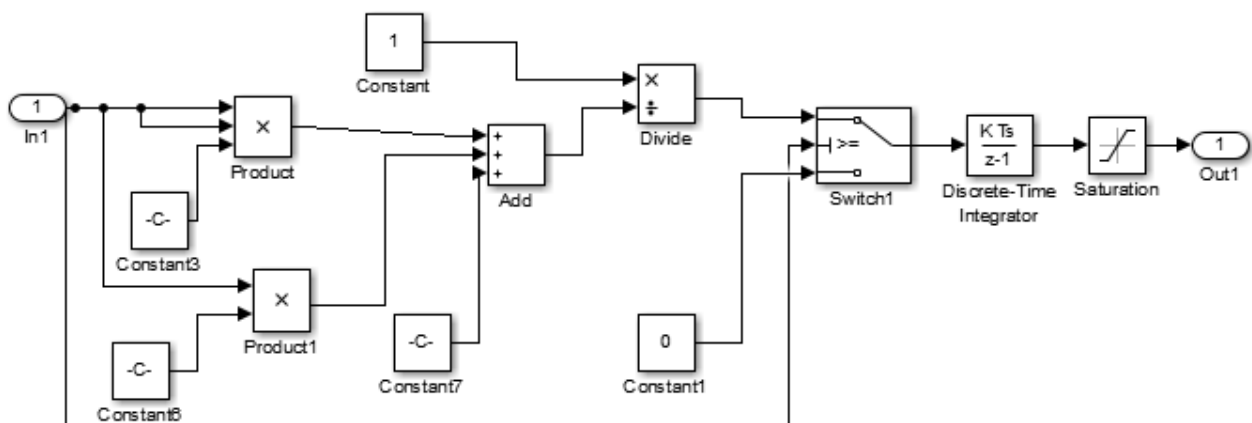


Рис. 4.3 - схема для розрахунку ступеня графітування

З результатів на рис. 4.4 видно, що при $t=55$ годин (210 ітерацій) заготовка набуває ступеня графітування $\beta = 0,98$.

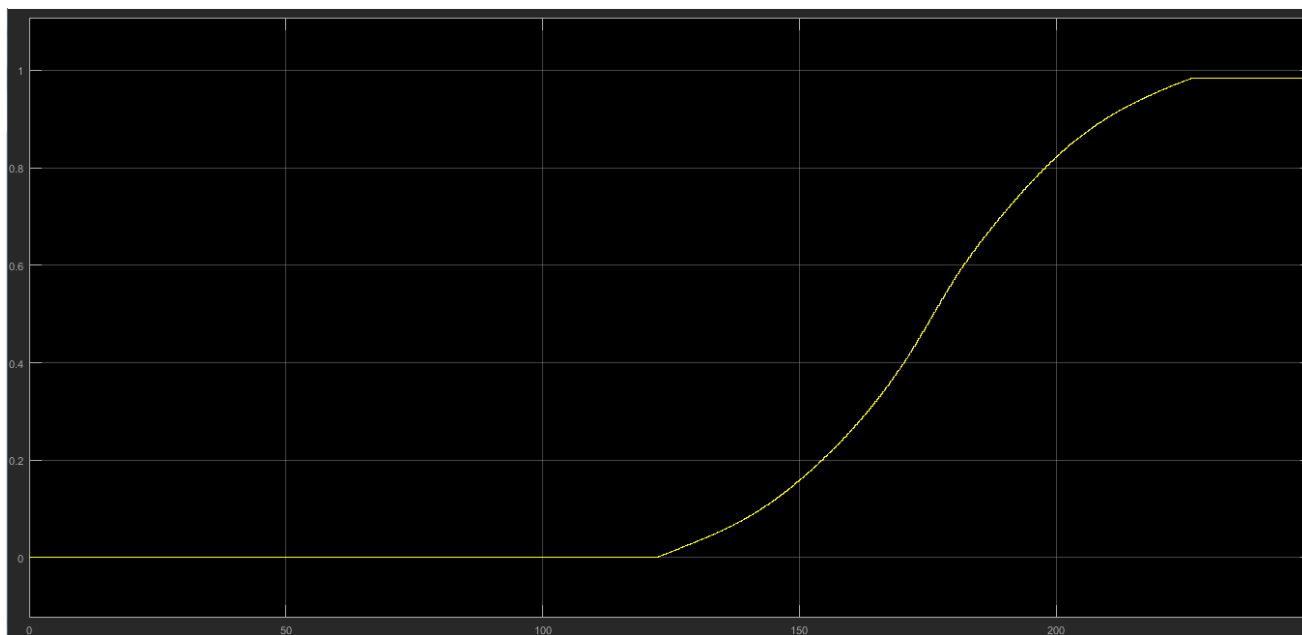


Рис. 4.4 - ступінь графітування при $t=55$ годин (210 ітерацій)

Після чого необхідно вимкнути подачу живлення саме в цей момент та дослідити на скільки зможе дографітуватись заготовка за рахунок акумуляції тепла пересипкою.

При вимкненні живлення на 55 годині (210 ітерацій) відбувається охолодження, проте до досягнення 2000°C продовжується графітування заготовки за рахунок закумульованого тепла.

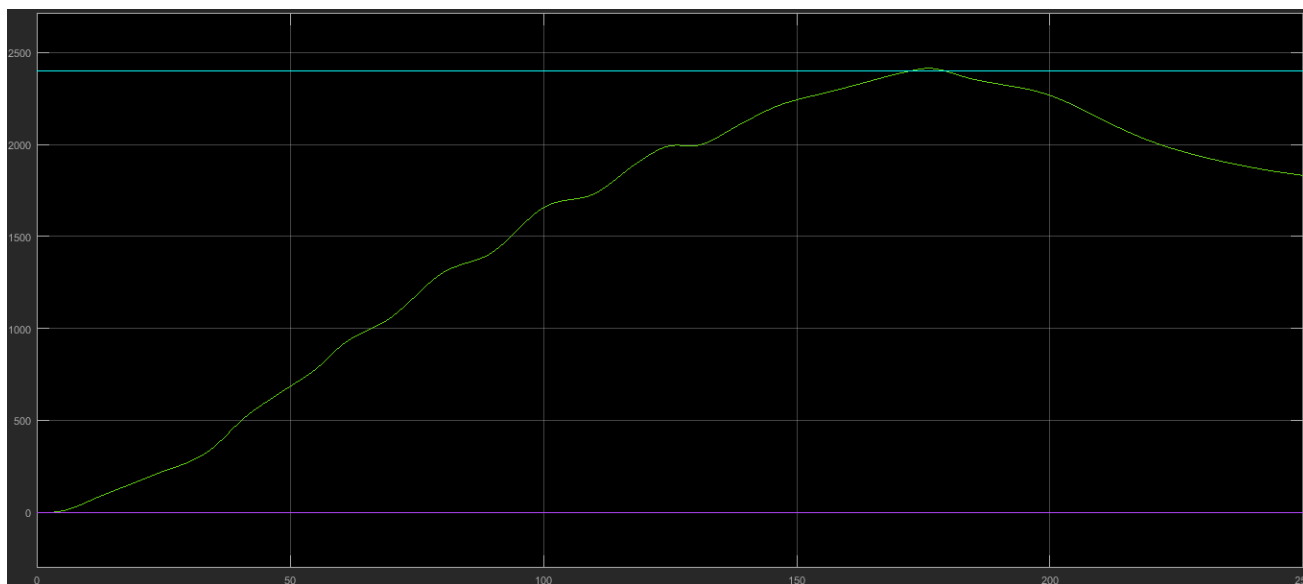


Рис. 4.5 - перехідна характеристики після вимкнення живлення

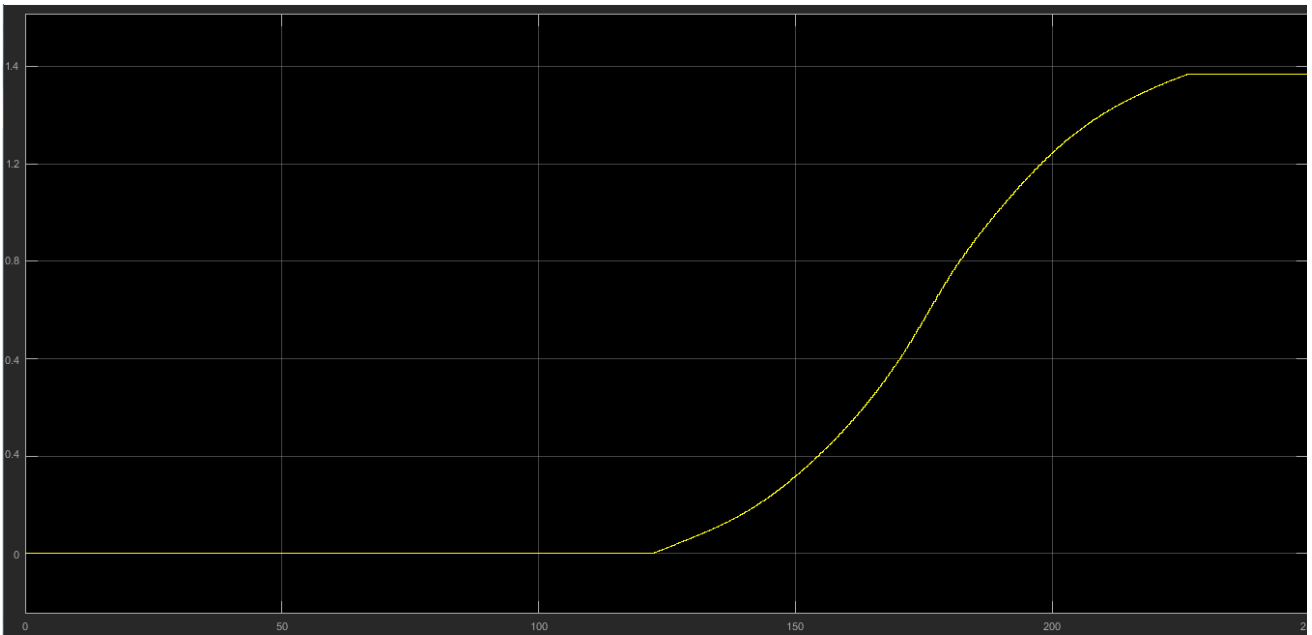


Рис. 4.6 - ступінь графітування після вимкнення живлення

Коли ми відключимо живлення на 47 годині (180 ітерації), заготовка дографітувалася до ступеня $\Delta\beta = 0,47$, що дозволяє казати, що при відключення живлення при 2200 °С ми можемо дографітувати заготовку на 47 %.

Проведені аналогічні досліди для температур від 2000 °С до 3000 °С, дозволяють побудувати залежність, що описує залежність дографітування до температури при вимкненні живлення. Описана вище залежність наведена на рис. 4.7.

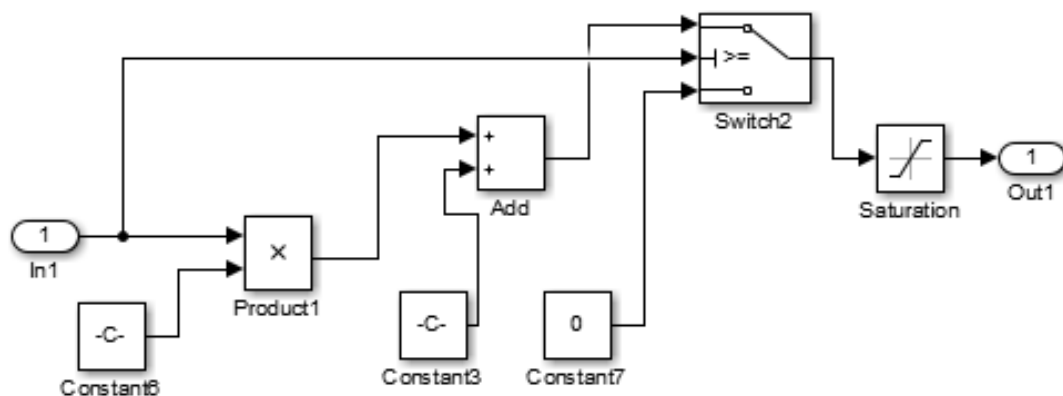


Рис. 4.7 - залежність дографітування до температури при вимкненні живлення реалізована в Simulink

Провівши вище роботу та отримавши результати, можна запропоувати систему керування, алгоритм роботи якої буде заключатися в наступному: розрахунок поточного значення ступеня графітування та дографітування в кожен момент часу, починаючи від 2000 °С, перевірка відповідності суми цих значень $\beta = 0,98$, відповідно, якщо сума дорівнює 0,98, то заготовка дографітується при поточному значенні температури, якщо вимкнути живлення.

Схема реалізації наведеного алгоритму :

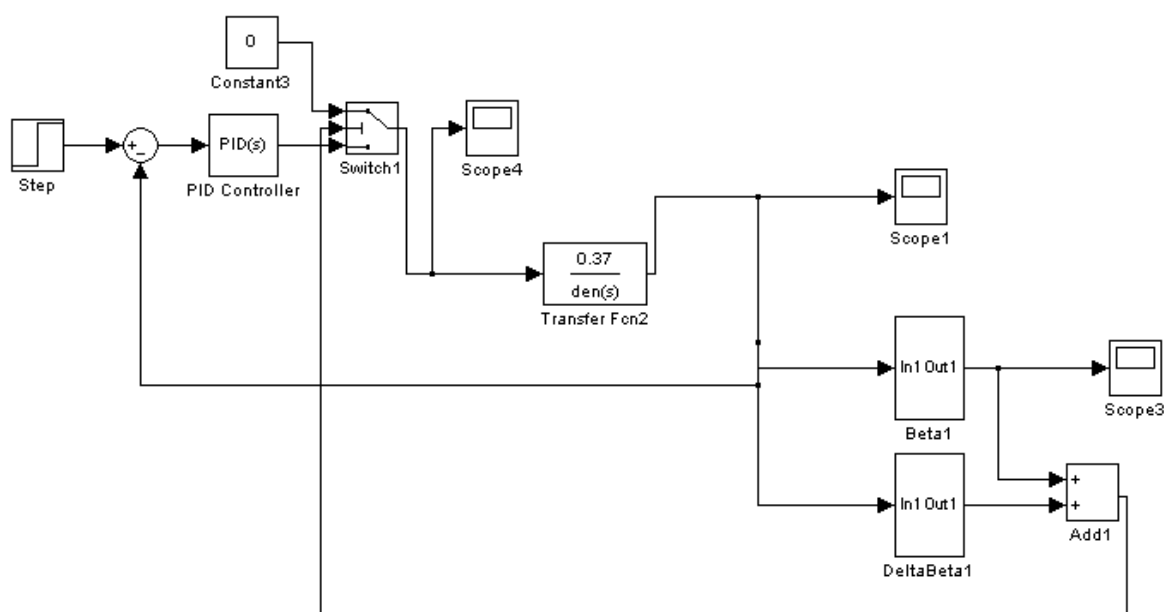


Рис. 4.8 - схема реалізованого алгоритму

Так як сума ступеня графітування і можливого дографітування становить близько 0,98 – можемо говорити про те, що ми можемо відключити живлення раніше, що і було реалізовано.

Очевидно, що даний алгоритм керування необхідно проводити по мінімальній температурі найхолоднішої заготовки, що забезпечить вихід на ступінь графітування даної та всіх інших, тепліших, заготовок, адже їх поточна температура вище, а тому і ступінь графітування та можливого дографітування вище.

Відвертим мінусом даної системи керування є її не оптимальність, позаяк можливий значний перепаду температур по заготовкам, а відповідно і значний перегрів всіх заготовок для досягнення відповідного ступення графітування

найхолоднішої. Це в свою чергу може призвести до недоцільного, по економічним показникам, підведення живлення.

Через це, пропонується перероблена система керування процесом графітування, що включає в себе наведений алгоритм попереднього відключення живлення та розрахунок економічної доцільності прогрівання кожної наступної заготовки.

Алгоритм описаної системи включає в себе: розрахунок ступеня графітування та дографітування в кожен наступний момент часу, починаючи від моменту досягнення 2000 °С, перевірку досягнення відповідного ступеня графітування по всім заготовкам, при досягненні однієї заготовки ступеня графітування 1 відбувається розрахунок економічної потреби графітування заготовок, що залишились. При потребі, відбувається продовження процесу, при недоцільності графітування - відключення живлення і дографітування відповідних заготовок, сума критерію графітування та дографітування яких $\beta = 0,98$.

Розрахунок економічної ефективності включає в себе: розрахунок витрат на електроенергію впродовж часу від початку кампанії до поточного моменту,

$$E = Q \times t \times P \quad (4.2)$$

де Q – кількість затраченої електроенергії (Вт /год), t – час, що затрачений на кампанію, P – ціна за електроенергію;

розрахунок прибутку:

$$S = N \times C \quad (4.2)$$

де N – кількість заготовок, для яких в даний момент часу сума критерію графітування та дографітування $\beta = 0,98$, C – ціна цих заготовок.

Наступний крок – розрахунок часу необхідного для дографітування заготовок, що залишились, в залежності від поточного ступеню графітування. З отриманого часу можна розрахувати необхідні витрати для забезпечення потрібної якості. На цьому ж етапі відбувається прийняття рішення щодо продовження графітування. Якщо витрати на дографітування тої чи іншої заготовки перевищують допустимі витрати - можна відключати живлення.

завантаженні, що дозволяє точніше розрахувати оптимальний час, коли вимкнути подачу електроенергії.

5. РОЗРОБКА ПРОГРАМНОГО ЗЕБЕЗПЕЧЕННЯ ДЛЯ КЕРУВАННЯ ПРОЧЕСМО ГРАФІТУВАННЯ У ПЕЧАХ АЧЕСОНА

5.1. Актуальність розробки програмного забезпечення

У час інформаційної епохи все більше набуває популярності створення програм для керування та демонстрації процесів. Розробка програмного забезпечення включає в себе багато стадій: проектування, програмування, тестування, впровадження і підтримку.

Проектування починається із формулювання вимог до програмного забезпечення і створення специфікацій — документів, у яких описані функції, що їх повинна виконувати програма. На наступному етапі створюється загальний дизайн програми: розбиття її на окремі блоки і визначення взаємодії між ними. На етапі безпосереднього програмування створюється текстовий код програми на одній чи декількох мовах програмування. Після компіляції коду, програмний продукт обов'язково проходить тестування, у процесі якого визначається відповідність продукту специфікаціям, знаходяться і виправляються помилки.

Перед впровадженням програмний продукт потребує документації — опису можливостей, посібників користувача, системи допомоги. Після впровадження програмного забезпечення, що для програмних продуктів вимагає маркетингу, системи дистрибуції, реклами тощо, програмне забезпечення потребує підтримки. Необхідність у підтримці виникає внаслідок швидкого розвитку комп'ютерів, що зумовлює необхідність взаємодії програмного продукту з іншими, новішими програмами і новою матеріальною базою. Часто підтримка нових можливостей забезпечується випуском нових версій програмного продукту.

Частиною програмного забезпечення є стандартні протоколи, які розробляються для узгодження програмних продуктів від різних виробників. Це потрібно для того, щоб, наприклад, електронний лист, надісланий через

електронну пошту з одного комп'ютера міг бути прочитаний на іншому комп'ютері зовсім іншою програмою та, навіть, з іншою операційною системою.

Користувач отримує програмне забезпечення разом із ліцензією, яка надає йому право використовувати програмний продукт за умови виконання положень ліцензування. Зазвичай, ці умови обмежують можливості користувача передавати програмний продукт іншим користувачам. Частина програмного забезпечення поставляється з вільною ліцензією (вільне програмне забезпечення). Такі ліцензії дозволяють розповсюджувати програмний продукт, а також модифікувати його.

Частина програмного забезпечення розповсюджується як безкоштовне. Існує також умовно безкоштовне або шароварне програмне забезпечення. У цьому випадку зазвичай користувач безкоштовно отримує демонстраційну версію програмного продукту з дещо обмеженими можливостями на певний випробувальний період, а після його закінчення зобов'язаний або придбати продукт, або видалити його.

У деяких країнах можливий патент на програмне забезпечення, однак процес і предмет патентування викликає численні дискусії.

5.2. Проектування програмного забезпечення для процесу графітування у печах Ачесона

Графітування – надзвичайно складний процес, що передбачає собою багато стадій виробництва, а також високі вимоги до кожної з них. Нас цікавить конкретний об'єкт – піч графітування. Мета створення програмного забезпечення – можливість відображення у режимі реального часу (або симуляції) всіх стадій графітування у печі Ачесона та забезпечити можливістю керувати процесом.

Піч графітування отримує на вхід напругу та силу струму, завдяки яким буде впливати на 9 вихідних параметрів: мінімальна температура нагріву, перепад температур і швидкість нагріву (все по 3 рядах). Отже, потрібно забезпечити виведення цих даних на екран персонального комп'ютера (далі – ПК). Окрім цього, потрібно додати можливості керування процесом різними

алгоритмами, оперуючись на різні методи. Додатково треба написати алгоритм подій при аварії на виробництві, також, буде корисно вивести якісь додаткові функції, наприклад, обрахунок коштів, яку буде затрачено на електроенергію, експорт даних для подальшого аналізу для розрахунку у *MS EXCEL*. Окремо, для оператора, який буде сидіти за комп'ютером, корисно вивести графіки, які показуватимуть зміни вихідних параметрів у часі.

5.3. Вибір середовища розробки програмного забезпечення

Перед тим, як розпочати розробку, необхідно визначити у якому саме пакеті найкраще це робити. На даний час, одним із найкращих програмних пакетів для програмування є середовище *C#.Net*.

На сьогоднішній момент мова програмування *C#* один з найпотужніших, що швидко розвиваються і затребуваних мов в ІТ-галузі. На даний момент на ньому пишуться найрізноманітніші програми: від невеликих десктопних програмок до великих веб-порталів і веб-сервісів, які обслуговують щодня мільйони користувачів.

У порівнянні з іншими мовами *C#* досить молодий, але в той же час він вже пройшов великий шлях. Перша версія мови вийшла разом з релізом *Microsoft Visual Studio .NET* в лютому 2002 року. Поточною версією мови є версія *C# 7.0*, яка вийшла в 7 березня 2017 року разом з *Visual Studio 2017*.

C# є мовою з *Cі*-подібним синтаксисом і близький в цьому відношенні до *C++* і *Java*. Тому, якщо ви знайомі з одним з цих мов, то опанувати *C#* буде легше.

C# є об'єктно-орієнтованим і в цьому плані багато перейняв у *Java* і *C++*. Наприклад, *C#* підтримує поліморфізм, успадкування, перевантаження операторів, статичну типізацію. Об'єктно-орієнтований підхід дозволяє вирішити завдання з побудови великих, але в той же час гнучких, масштабованих і розширюваних додатків. *C#* продовжує активно розвиватися, і з кожною новою версією з'являється все більше цікавих функціональностей, як, наприклад, лямбда, динамічне зв'язування, асинхронні методи і т.д.

Коли говорять *C#*, нерідко мають на увазі технології платформи *.NET* (*WPF*, *ASP.NET*). І, навпаки, коли говорять *.NET*, нерідко мають на увазі *C#*. Однак, хоча ці поняття пов'язані, ототожнювати їх невірно. Мова *C#* був створений спеціально для роботи з фреймворком *.NET*, проте саме поняття *.NET* дещо ширше.

Якось Білл Гейтс сказав, що платформа *.NET* - це найкраще, що створила компанія Microsoft. Можливо, він мав рацію. Фреймворк *.NET* представляє потужну платформу для створення додатків. Можна виділити наступні її основні риси:

Підтримка декількох мов. Основою платформи є загальномовне середовище виконання *Common Language Runtime (CLR)*, завдяки чому *.NET* підтримує кілька мов: поряд з *C#* це також *VB.NET*, *C++*, *F#*, а також різні діалекти інших мов, прив'язані до *.NET*, наприклад, *Delphi .NET*. При компіляції код на будь-якому з цих мов компілюється в збірку спільною мовою *CIL (Common Intermediate Language)* - свого роду асемблер платформи *.NET*. Тому ми можемо зробити окремі модулі однієї програми на окремих мовах.

Кросплатформеність. *.NET* переносять платформою (з деякими обмеженнями). Наприклад, остання версія платформи на даний момент *.NET Framework* підтримується на більшості сучасних ОС *Windows (Windows 10 / 8.1 / 8 / 7 / Vista)*. А завдяки проекту *Mono* можна створювати додатки, які будуть працювати і на інших ОС сімейства *Linux*, в тому числі на мобільних платформах *Android* і *iOS*.

Потужна бібліотека класів. *.NET* представляє єдину для всіх підтримуваних мов бібліотеку класів. І яке б додаток ми не збиралися писати на *C#* - текстовий редактор, чат або складний веб-сайт - так чи інакше ми задіємо бібліотеку класів *.NET*.

Різноманітність технологій. Загальномовне середовище виконання *CLR* і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при побудові тих чи інших додатків. Наприклад, для роботи з базами даних в цьому стеку технологій призначена технологія *ADO.NET*. Для

побудови графічних додатків з багатим насиченим інтерфейсом - технологія *WPF*. Для створення веб-сайтів - *ASP.NET* і т.д.

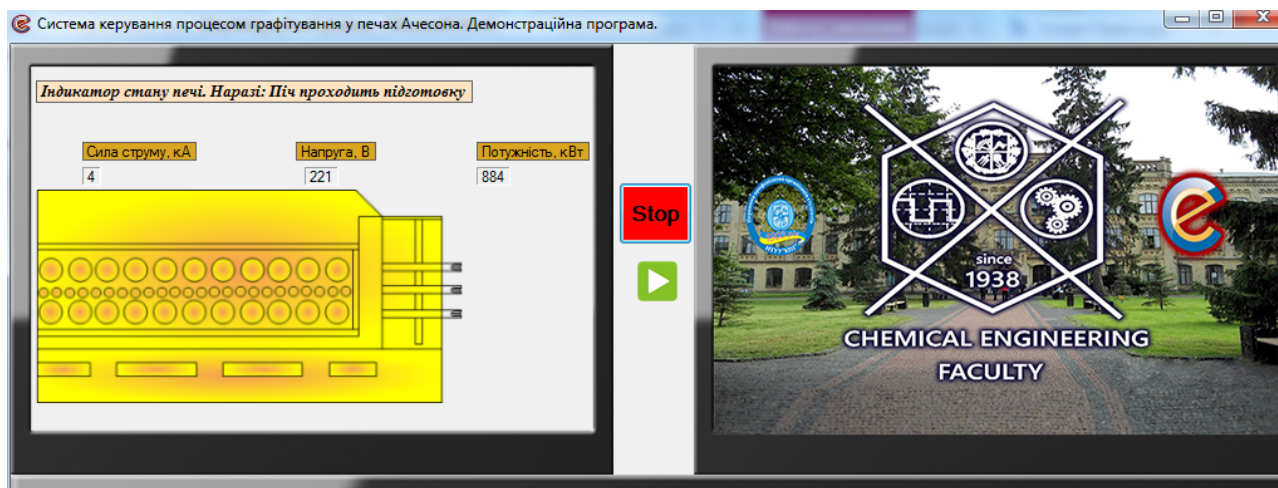
Також ще слід відзначити таку особливість мови *C#* і фреймворка *.NET*, як автоматичне прибирання сміття. А це означає, що нам в більшості випадків не доведеться, на відміну від *C++*, піклуватися про звільнення пам'яті. Вищезазначена загальномовного середовища *CLR* сама викличе збирач сміття і очистить пам'ять.

5.4. Розробка програмного забезпечення

Вибравши середовищем розробки платформу *C# .Net*, будемо спочатку проектувати програму завдяки пакету *Windows Forms*. Зробимо 3 екрани, на яких будемо розташовувати необхідні нам інструменти та дані. На першому екрані відобразимо саму піч, дані про напругу, силу струму та потужність, що наразі підведена до печі, та індикатор події, щоб розуміти на якій стадії ми знаходимося наразі. Другий екран буде містити органи керування – кнопки запуску та зупинки печі, режими роботи печі, кнопка експорту даних до *MS EXCEL*. Також, там потрібно зробити індикатори, на яких буде відображатися дані про значення сили струму, напруги, потужності, витрати коштів на електроенергію. Окрім цього, там буде розташована кнопка аварія, яка дозволить вказати про аварію на виробництві та заблокувати всі події для уникнення вибуху та інших не бажаних наслідків. Третій екран буде присвячений індикації вхідних та вихідних параметрів, себто там буде відображатися зміна мінімальної температури, швидкість нагріву та перепад температур по 3 рядах, а також зміна напруги і сили струму.



Рис. 5.1. Вигляд програми при запуску

Рис. 5.2. Підготовка печі, що відбувається після натиску кнопки *Start*

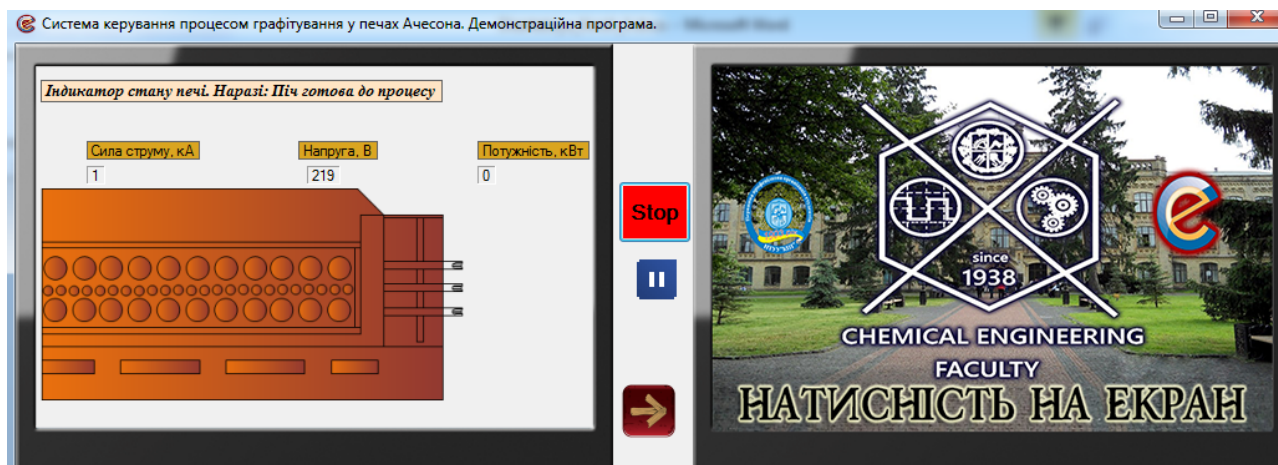


Рис. 5.3. Завершення підготовки печі

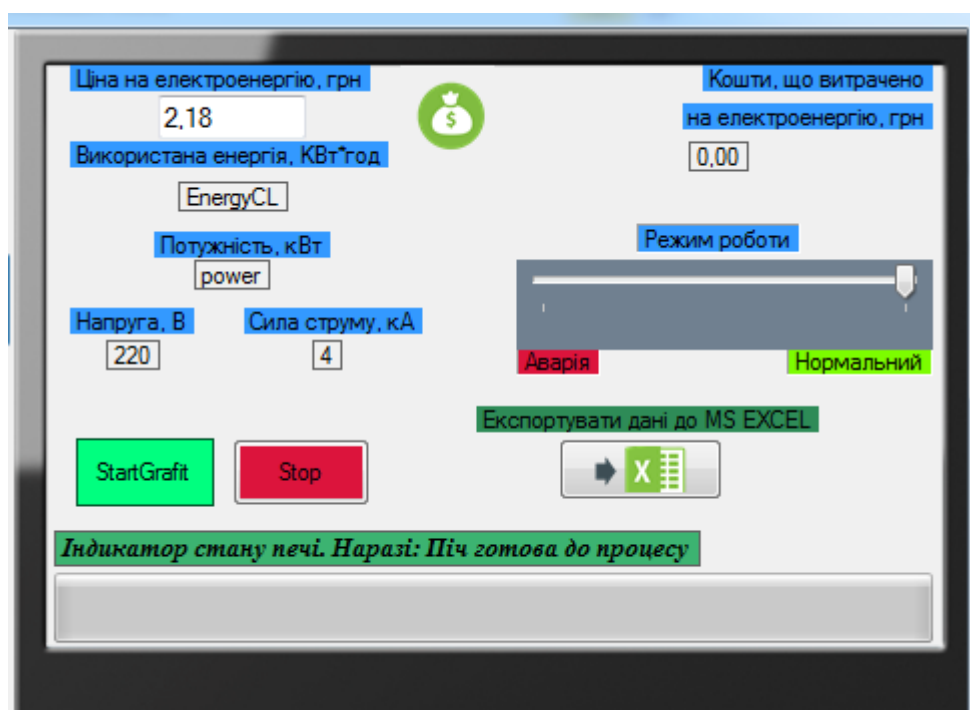


Рис. 5.4. Відкриття доступу до панелі керування початком графітування

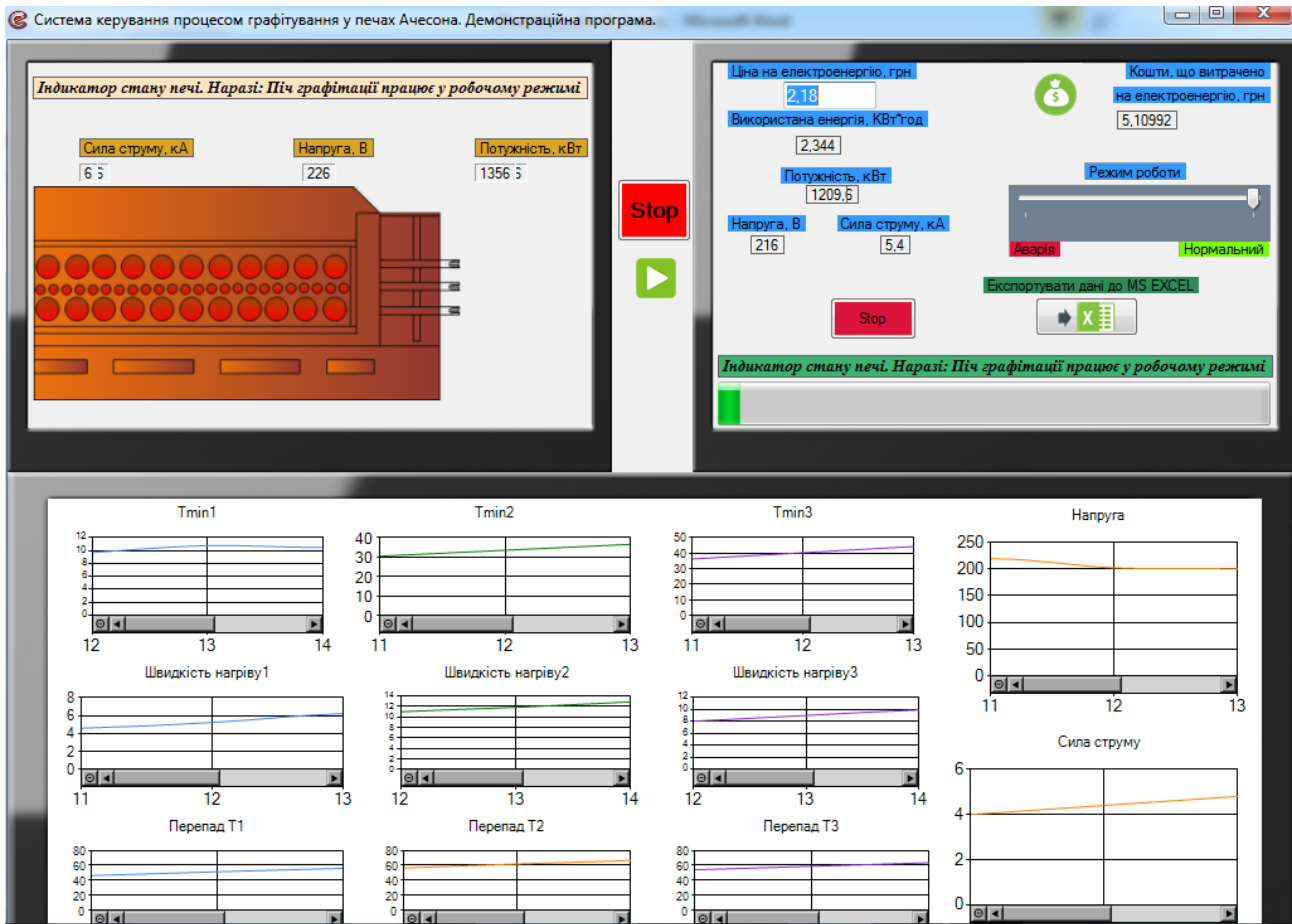


Рис. 5.5. Старт процесу графітування

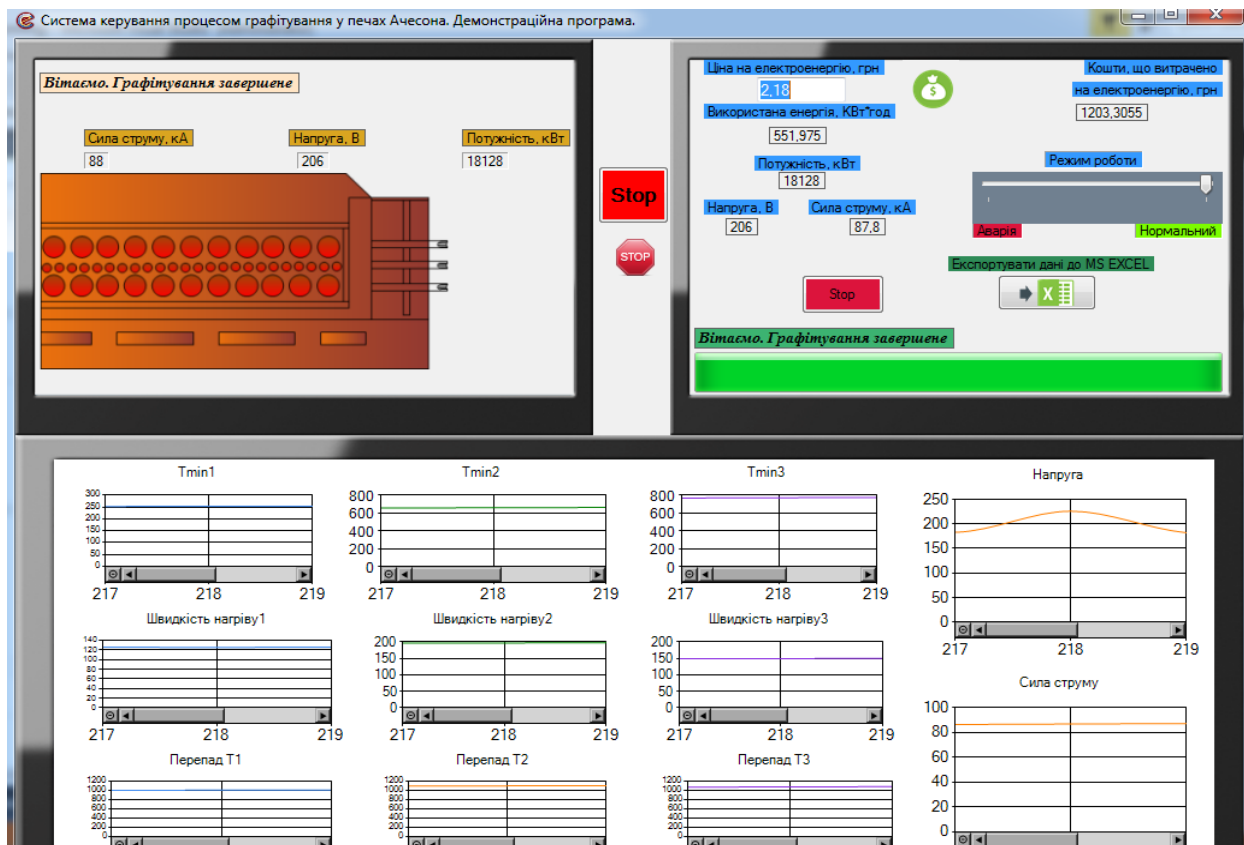


Рис. 5.6. Успішне закінчення процесу графітування

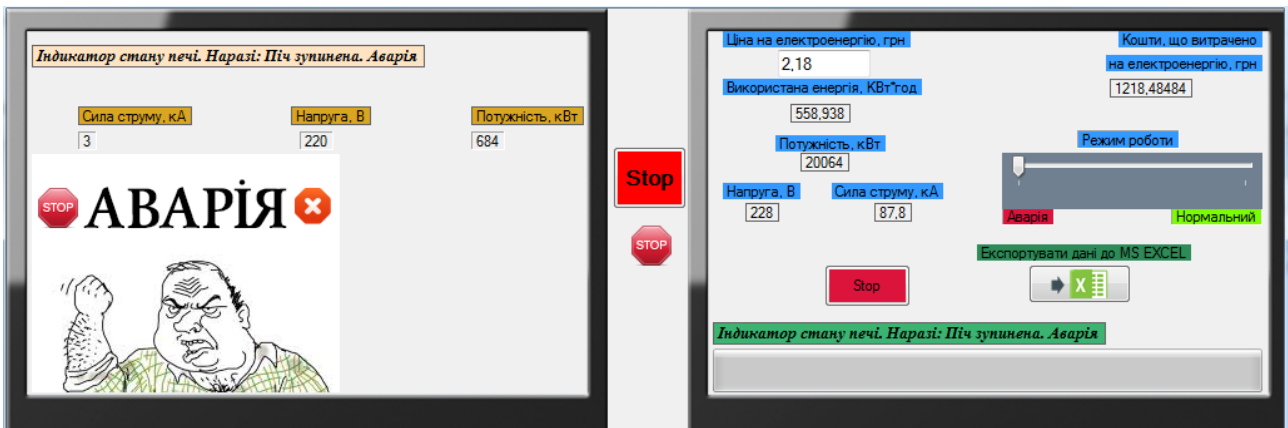


Рис. 5.7. Режим аварії

ExcelFile.xlsx - Microsoft Excel

Файл

Главная

Вставка

Разметка страницы

Формулы

Данные

Рецензирование

Вид

Вставить

Буфер обмена

Calibri

11

Ж

Б

У

Рис. 5.8. Експортовані дані до MS EXCEL

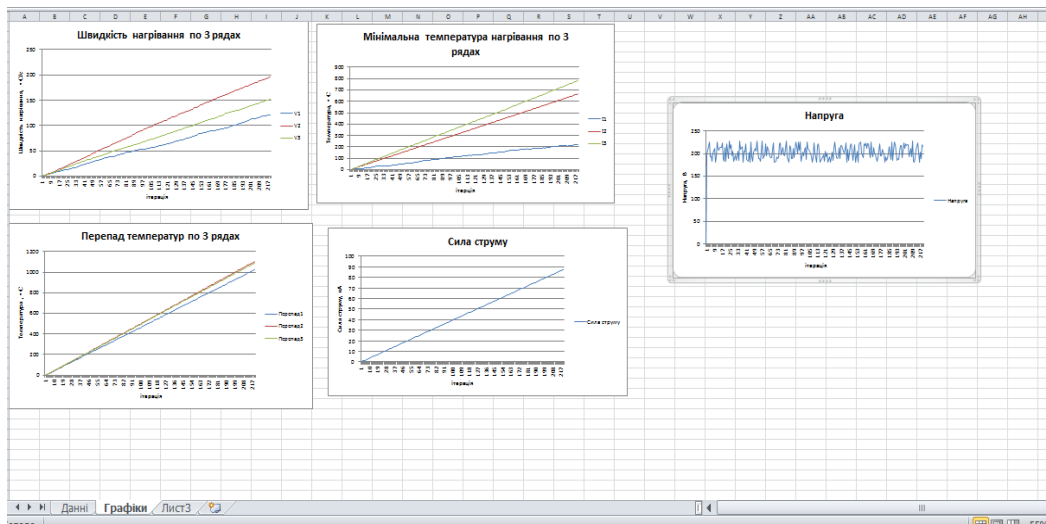


Рис. 5.8. Графіки експортованих даних до MS EXCEL

Лістинг програмного забезпечення даної програми можна побачити у додатку 1.

5.5. Розробка програмного забезпечення для віддаленого доступу

Розробивши програмне забезпечення завдяки *Windows Forms*, протестувавши його, можна перейти до наступного етапу – можливості надання віддаленого доступу до програмного забезпечення керування ним на відстані через локальну мережу або мережу інтернет. Для цього потрібно розробити програмне забезпечення у *WPF Visual Studio 2017*. Потім купити *ASP.NET* хостинг та закинути програмний код. Відповідно, якщо все вірно синхронізувати можна з іншого кутка світу керувати даним процесом.

Windows Presentation Foundation (WPF) - це система наступного покоління для побудови клієнтських додатків *Windows* з візуально привабливими можливостями взаємодії з користувачем. За допомогою *WPF* можна створювати широкий спектр як автономних, так і розміщених в браузері додатків.

В основі *WPF* лежить векторна система візуалізації, яка не залежить від дозволу і створена з розрахунком на можливості сучасного графічного обладнання. *WPF* розширює базову систему повним набором функцій розробки додатків, в тому числі *Extensible Application Markup Language (XAML)*, елементами управління, прив'язкою даних, макетом, 2-D- і 3-D-графікою, анімацією, стилями, шаблонами, документами, мультимедіа, текстом і оформленням. *WPF* входить до складу *Microsoft .NET Framework* і дозволяє створювати додатки, що включають інші елементи бібліотеки класів *.NET Framework*.

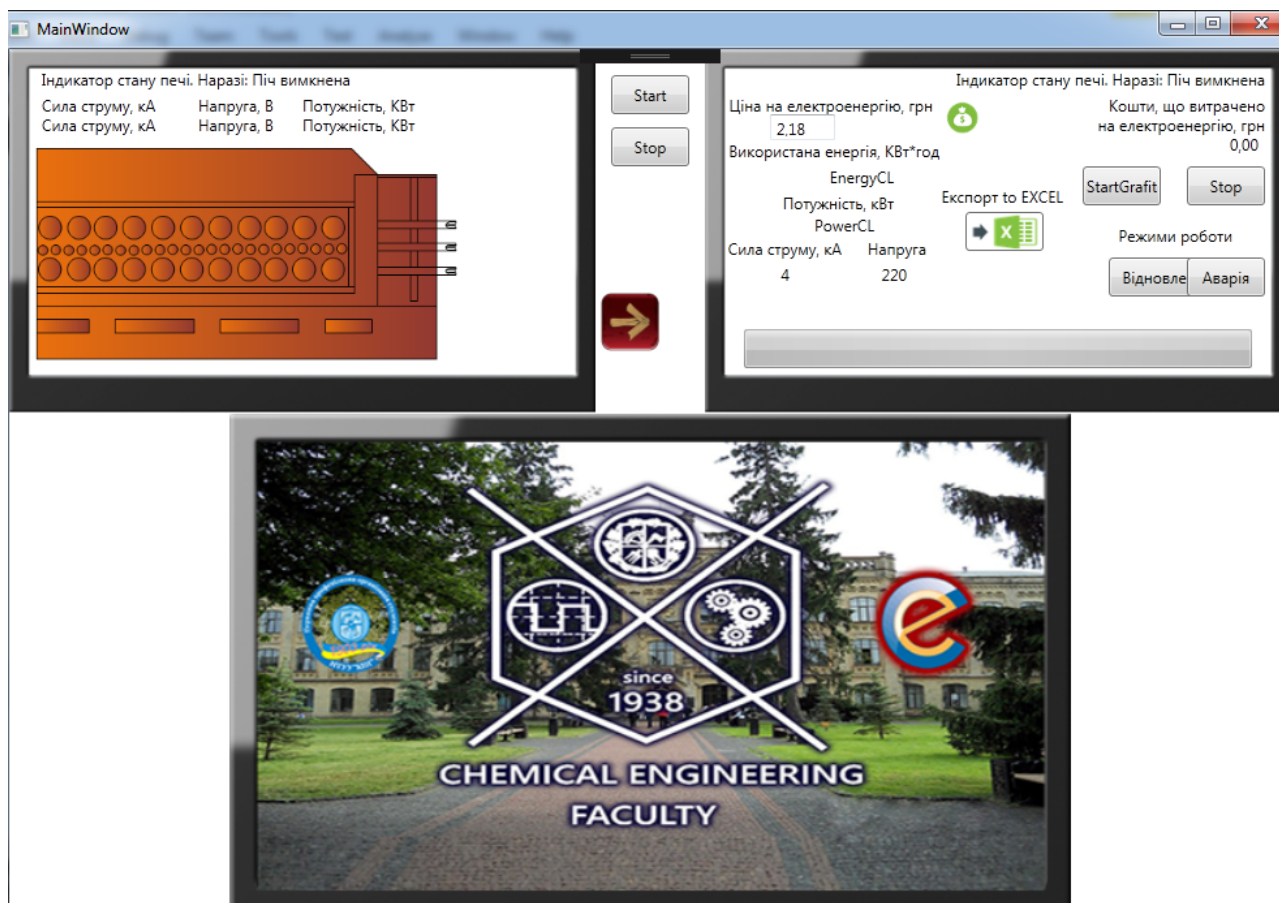


Рис. 5.9. Запуск програми у WPF

Лістинг програмного забезпечення даної програми можна побачити у додатку 2.

6. СТВОРЕННЯ СТАРТАП ПРОЕКТУ

6.1 Опис ідеї стартап-проекту

Важко уявити сучасну промисловість без використання графітових електродів. Практично кожна галузь так чи інакше потребує електродну продукцію. Найбільше їх потребують чорна та кольорова металургія, гальванопластика, тяжке машинобудування, хімічна промисловість тощо.

Особливе місце серед електродної продукції посідає графітована електродна продукція, перш за все через високі показники якості, такі, як висока міцність, теплопровідність, низький питомий опір тощо.

Планується розробити та спроектувати систему автоматизованого керування процесом графітування при виробництві електродної продукції, що зможе оптимізувати енергозатрати та підвищити якість продукції.

Електродна продукція є неоціненною у своєму практичному застосуванні. Це зумовлено унікальними фізико-механічними властивостями, що дозволяє використовувати її в різних галузях промисловості у виробництві алюмінію, фосфору, хлору і т.д. Всі технологічні операції мають вагоме значення для забезпечення високого виходу якісної електродної продукції. Важливо організувати прийом і закрите зберігання сировини, що має різні властивості, а також дотримання режимів заміщення, охолодження і пресування електродної маси.

Основним фактором якості графітованої продукції є її міцність, яка залежить від: вологості заготовки перед входом у піч (від 4 до 16%), температурою у печі, часу випікання, швидкості нагрівання та перепаду температур. У реальному часі важко відслідкувати і підібрати найоптимальніші варіанти цих параметрів для високої якості продукції. Саме тому, важливо спроектувати автоматичну систему керування, яка буде знаходити оптимальні умови для керування процесом.

Таблиця 6.1 – Опис ідеї стартап-проекту

<i>Напрямки застосування</i>	<i>Вигоди для користувача</i>
Графітування у виробництві графітових електродів	<ul style="list-style-type: none"> - Зменшення затрат енергії та інших ресурсів на виробництві - Висока якість - Зменшення собівартості продукції

6.2 Визначення сильних, слабких та нейтральних характеристик ідеї проекту

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик проекту

№ п/ п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	Конкурент 1	Конкурент 2			
1.	Час графітування	70-100 годин	100-140 годин	100-140 годин			+
2.	Якість продукту графітування	висока	середня	середня			+
3.	Витрати електроенергії менші на	8 %	3 %	Не менші			+
4.	Температура графітування	3000 °C	2800 °C	3000 °C		+	

6.3 Технологічна здійсненність ідеї проекту

Головною ідеєю зменшення затрат та покращення якості продукції – є зменшення впливу людини на проходження процесу майже до нуля. Пропонується використати розроблену систему автоматичного керування процесом, котра дозволить керувати всім тонкощами процесу графітування завдяки ПК. Це буде можливо навіть на відстані завдяки мережевим з'єднанням, тобто людина може бути у іншому кутку світу, а процес буде іти безвідмовно і у

випадку несправностей він зупиниться, спрацює аварійна сигналізація і все закінчиться без проблемних наслідків для робочих на заводі. Також, варто відмітити, що завдяки цьому програмному забезпеченню можна буде скоротити штат працівників, що обслуговують піч, а це значно зменшить собівартість продукції. У програмному продукті буде вмонтовано регулятори різних типів, які будуть автоматично підбиратися, але можливо також буде вибрати регулятор вручну (якщо знати особливості процесу безпосередньо на місці). Система обмежень програмного комплексу зможе пришвидшити процес збільшення температури розігріву печі до надвисоких температур.

Окрім вищевказаного варто додати деякі технічні нюанси. У процесі графітування в печах непрямого нагріву Ачесона для запобігання впливу вологи на ріст температури керна запропоновано встановлювати захисні щити з волокнистого теплоізоляційного матеріалу вздовж бічних поверхонь керна. Для забезпечення рівномірного нагріву заготовок по рядах запропоновано укладати заготовки таким чином, щоб у верхньому ряді їх було на одну більше, ніж у нижньому, що виключає завищені температури та знижує теплові втрати.

6.4 Характеристика потенційних клієнтів

Зважаючи на сучасний економічний, соціальний стан нашої держави, необхідність економити власні кошти мають багато приватних та державних підприємств, окремих регіонів та груп населення (наприклад, чорна та кольорова металургія, гальванопластика, тяжке машинобудування, хімічна промисловість).

Таблиця 6.3 – Характеристика потенційних клієнтів

<i>n/n</i>	<i>Потреба, що формує ринок</i>	<i>Цільова аудиторія (цільові сегменти ринку)</i>	<i>Відмінності у поведінці різних потенційних цільових груп клієнтів</i>	<i>Вимоги споживачів до товару</i>
1	2	3	4	5
	Базова потреба, яку задовольняє товар (згідно концепції потенційного товару)	Визначити потенційні цільові групи клієнтів, що можуть бути зацікавлені у задоволенні значеної потреби	Вписати фактори, що формують поведінку клієнта (стандарты, технічні регламенти, інші фактори цінового та нецінового характеру)	<ul style="list-style-type: none"> - до продукції - до компанії -постачальника

			та особливості купівлі та експлуатації товару	
--	--	--	--	--

Продовження таблиці 6.3.

1	2	3	4	5
1.	Графітування Графітових електродів	Чорна та кольорова мета- лургія, галь- ванопластика, тяжке машинобу- дування, хімічна промисловість	ДСТУ, ГОСТ	Достатня економія електрое- нергії при виробництві згідно ДСТУ та правдивих резуль- татів постійний контроль

6.5 Фактори загроз

Загрози існують скрізь і при застосуванні даної технології також є загрози. Для попередження таких ситуацій необхідно встановлювати якісні та сучасні системи контролю та спостережень, а також працювати на очисних установках повинні висококваліфіковані фахівці. Також, на виході після очищення води за даною технологією необхідно виконувати регулярний контроль та аналіз очищених вод, щоб якомога швидше усунути несправність або помилки, у разі їх виникнення.

Таблиця 6.4 – Фактори загроз

№п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Недостатня економія, внаслідок неправильного використання	Низька якість керування процесом	Ремонт або налаштування системи керування
2.	Недостатня якість готового продукту, внаслідок виходу із ладу необхідного обладнання	Вихід із ладу обладнання в печі Ачесона	Ремонт або налагодження обладнання
3.	Недостатня температура для графітування внаслідок застарілого обладнання	Готова продукція не високої якості	Модернізація обладнання або ж заміна на нове

6.6. Фактори можливостей

Таблиця 5.5 – Фактори можливостей

№ n/n	Фактор	Зміст можливості	Можлива реакція компанії
1.	Застосування печі графітування лише за інструкцією	Економія електроенергії та уникнення несправності електромережі	Збільшення об'єму виготовленої продукції
2.	Вчасна модернізація обладнання	Знижує загрозу виготовлення продукції низької якості	Збільшення об'єму виготовленої продукції
3.	Встановлення додаткових термопар, для виміру температури в печі	Знижує загрозу роботи печі на неправильній температурі	Збільшення об'єму виготовленої продукції

6.7. Ступеневий аналіз конкуренції на ринку

Таблиця 6.6 - Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	2	3
1. Тип конкуренції : Чиста	Ніхто не може вплинути на ситуацію на ринку безпосередньо. Лише інновації та вигідні пропозиції	Система знижок, програми лояльності. Довготривалі контракти, зниження ціни. Випробувальні терміни.
2. За рівнем конкурентної боротьби : національна (згодом і міжнародна)	Треба орієнтуватися спочатку на національний ринок, адже імені і компанії ще немає. Потрібно працювати максимально з усіма фірмами і розширювати ім'я.	Розсилати електронною поштою листи та шукати особистої зустрічі з підприємцями
3. За галузевою ознакою : внутрішньогалузева.	Автоматизовані системи керування по факту мають собою різницю лиш у якості, ціні і функціях.	Робити якісні АСК, які будуть робити товар вищої якості з використанням менше ресурсів.

Продовження таблиці 6.6.

1	2	3
4. Конкуренція за видами то-варів: товарно-видова	Ми пропонуємо АСК з різними властивостями в залежності від потреби покупця.	Запропонувати або широкий спектр із дуже потужними властивостями
5. За характером конкурент-них переваг: нецінова	Наша продукція матиме трішки вищу ціну за середньоринкову, основний акцент варто зробити на якість продукції	Пропозиція з випробувальним терміном. Вказівка на економії у часі, ресурсі та якості вихідної продукції.
6. За інтенсивністю: марочна	Потрібно зробити бренд із нашого продукту, щоб його можна було легко впізнати	Створення логотипу, нанесення його (або назви) на продукт. Реклама в інтернеті.

6.8. Аналіз конкуренції в галузі за М. Портером

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі.

Таблиця 6.7 - Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
1	2	3	4	5	6
	«Смарт Сервіс» (СС) «Вотум» (В) «Мікрол» (М)	Нові компанії, які починають розвиватися і впроваджують інновації. Надалі, при розширенні ним може стати «ABB» та «Siemens»	Сила у тому, що без АСК наразі виробництва не працюють. Можна поставити АСК інших виробників, але прибуток знизиться.	Клієнти можуть вимагати зниження ціни за рекламу і використання продукції протягом тривалого часу. Також вони можуть шукати щастя у інших фірмах.	Інші фірми виробляють схожу продукцію. Але є різні рівні цін та якості.

Продовження таблиці 6.7.

1	2	3	4	5	6
Висновки:	Основна перевага – ціна і досвід. Але враховуючи потенціал, можна заповнити їхній ринок	Є можливість входу в ринок. Конкуренти уже змогли стати брендами, але є можливість задавити їх завдяки вигідним пропозиціям	Диктують. Співвідношення якості/ціна/обсяг	Можуть наполягати на здешевленні.	Не можна завищити ціну.

6.9. Обґрунтування факторів конкурентоспроможності

Після всіх аналізів визначається та обґрунтовується перелік факторів конкурентоспроможності.

Таблиця 6.8 - Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1.	Висока якість	Чим вища якість продукту – тим більше буде прибутку.
2.	Доступна ціна	Ціна трішки вища середньої, але дає можливість себе окупити дуже швидко.
3.	Можливість надати випробувальний період	Клієнт має можливість сам переконатися у ефективності.
4.	Економія ресурсів при виготовленні	Економія ресурсів призводить до збільшення прибутку

З таблиці бачимо, що фактори конкурентоспроможності суттєві та мають великий позитивний внесок при запровадженні технології графітування за допомогою печі Ачесона.

6.10. Порівняльний аналіз сильних та слабких сторін «Процесу графітації за допомогою печі Ачесона»

За визначеними факторами конкурентоспроможності (табл.6.9) проводиться аналіз сильних та слабких сторін стартап-проекту (табл. 6.10).

Таблиця 6.9 - Порівняльний аналіз сильних та слабких сторін «назва проекту»

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з “КАР-Company”						
			-3	-2	-1	0	+1	+2	+3
1.	Висока якість	18		М	СС	В			
2.	Доступна ціна	10					В	СС	М
3.	Можливість надати випробувальний період	20	В, СС	М					
4.	Економія ресурсів при виготовленні	18		СС	М		В		

6.11. SWOT – аналіз стартап-проекту

Таблиця 6.10 - SWOT- аналіз стартап-проекту

<i>Сильні сторони</i>	<i>Слабкі сторони</i>
<ul style="list-style-type: none"> - Якісна продукція - Контракти з ВНЗ 	<ul style="list-style-type: none"> - Ціна вища ніж у конкурентів - Конкуренти, які створюють продукцію новітніми технологіями
<i>Можливості</i>	<i>Загрози</i>
<ul style="list-style-type: none"> - Заощадження на зарплаті - Навчати «своїх» працівників після ВНЗ 	<ul style="list-style-type: none"> - Люди не будуть купляти товар - Зниження якості роботи - Збільшення часу виготовлення продукції

Управлінська проблема:

Знайти можливість випускати продукцію такої ж якості за ринковою ціною та якісною навчити молодий персонал.

Таблиця 6.11 - Слабкі та сильні сторони альтернативних шляхів

Альтернативи	Слабкі сторони	Сильні сторони
1. Зробити систему знижок	Зменшення прибуток	Може урівняти ціни
2. Проводити тренінги	Затрати часу та коштів	Підвищення кваліфікації персоналу

Найоптимальніший шлях вирішення управлінської проблеми є – розробка системи знижок на товар та проведення тренінгів для молодого персоналу.

6.12. Вибір цільових груп потенційних споживачів

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (табл.6.12).

Таблиця 6.12 - Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Підприємства, що спеціалізуються на виготовленні електродної продукції	80%	65%	Середня конкуренція. Зазвичай компанії звертаються до не спеціалізованих фірм	Враховуючи те, що на цей напрям буде зроблений акцент – то це дасть нам перевагу
2.	Автоматизовані виробництва різних профілів	50%	45-50%	Велика конкуренція, особливо за рахунок використання іноземних компаній. Акцент можна робити на	Важко, адже нова компанія. Першозаявлений профіль – системи керування для вузької спеціалізації,
Які цільові групи обрано: 1. Виробництва, що спеціалізуються на виготовленні електродної продукції					

6.13. Визначення базової стратегії розвитку

Результатами аналізу потенційних груп споживачів (сегментів) ми обрали цільові групи, для яких будемо пропонувати свою технологію графітування та визначили стратегію охоплення ринку:

- стратегію диференційованого маркетингу, тому що працюємо із кількома сегментами, розробляючи для них окремо програми ринкового впливу. Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (табл.6.13).

Таблиця 6.13 - Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Зниження ціни до середньоринкових	Пропозиція із випробувальним періодом за символічну ціну.	1. Висока якість 2. Низька ціна 3. Регульований термін контракту і легке його розірвання	Стратегія лідерства по витратах
2.	Заклучення не довготривалих контрактів	Надання компаніям книгу відгуків від інших фірм; Відеозаписи.	1. Можливість легко розірвати контракт. 2. Безкоштовне надання своїх фахівців для обслуговування.	Стратегія диференціації

6.14. Визначення базової стратегії конкурентної поведінки

Наступним кроком є вибір стратегії конкурентної поведінки (табл.6.14).

Таблиця 6.14 - Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Ні	Обов'язково. Потрібно показувати свої переважні риси над іншими. Зараз важко знайти такі фірми, які ще не мають своїх постачальників. Хіба фірми які розвиваються і на початковому етапі знаходяться.	Якщо при базовій стратегії не вдасться заповнити ринок (а скоріш за все так і буде), то будемо брати конкурентну ціну. Обов'язково слідувати за нововведеннями і новими розробками та конкурентними пропозиціями інших компаній та аналізувати. Дещо удосконалювати і переробляти під себе.	Стратегія розширення первинного попиту, з подальшим переходом до наступальної стратегії.

6.15. Визначення стратегії позиціонування

Таблиця 6.15 - Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)
1.	1. Платоспроможність. 2. Максимально тривалий контракт.	Стратегія диференціації	1. Продукція вищої якості за конкурентну 2. Можливість економити на ресурсах 3. Можливість випробувального періоду за символічну ціну. 4. Програми лояльності за тривалі контракти	1. Висока якість. 2. Тривалі контракти. 3. Програма лояльності.

6.16. Визначення ключових переваг концепції потенційного товару

Таблиця 6.16 - Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі,що потрібно створити)
1.	Продукція актуальна. Наразі висока актуальність	Збільшення чистого прибутку. Покращення якості продукту	Якість продукції вища ніж у конкурентів. Дозволяє зекономити ресурси

6.17. Визначення меж встановлення ціни

Таблиця 6.17 - Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари - аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	відсутні	15-25 тис. грн	50-100 тис. грн	Верхня: 30 тис. грн Нижня: 23 тис. грн

6.18. Формування системи збуту

Таблиця 6.18 - Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Цільові клієнти хочуть мати систему, яка буде економити ресурси та покращувати якість своєї продукції. Вони будуть шукати у ціні щось середнє пропорційне.	Доставка Тест Гарантія Ремонт Допомога введення у експлуатацію	Нульовий рівень. Наша компанія сама встановлює і проектує датчики. Деякі з них закуповують, деякі модифікуються. До споживача доходить продукція з усіма необхідними компонентами.	Заклучення контрактів про поставки в певні терміни із щорічним нашим технічним обслуговуванням АСК

6.19. Концепція маркетингових комунікацій

Таблиця 6.19 - Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного відомлення	Концепція рекламного звернення
1.	Бажання заключити контракт на низьку ціну на невеликий термін.	Все спілкування і обговорення вживу та по телефону. Можлива комунікація у соціальних мережах та по електронній пошті. Замовлення приймається офіційно лише після підписання паперів.	Якість залежить від ціни; Тривале партнерство = лояльності ціни; Вчасно і економно.	Показати переваги продукції і можливий виграш у прибутку компанії. +Зручність обслуговування та вчасні поставки.	Офіційний стиль. Додавати факти, графіки. Шукати зустрічей. Пропозиції випробувального терміну для порівняння

6.20. Опис трьох рівнів моделі товару

Таблиця 6.20 - Опис трьох рівнів моделі товару

Обрана альтернатива розвитку проекту	Сутність та складові		
I. Товар за задумом	Автоматизована системи виробництва		
II. Товар у реальному виконанні	Властивості/характеристики	м ²	Вр/Тх /Тл/Е/Ор
	Система керування ,яка підключається до печі Ачесона (або іншого пристрою, якщо це вказано у замовленні) та побічних пристроїв, а також до комп'ютера та засобів керування. По факту це система датчиків та з'єднувальних шнурів, які вмонтовуються на потрібне місце (вказано у інструкції)	1x1	20 тис. грн Можна підключити на будь-якому зручному місці для обслуговуючого персоналу. Захищені відпотрапляння води та інших речовин. U=360V; та U=220V. I=20mA;
	Якість: відповідає Європейським нормам та ДСТУ. Тестування проводиться на перед відправкою ,а також на місці.		
	Пакування – картонова коробка із спеціальними захисними матеріалами всередині для уникання деформацій чи пошкодження		
	Марка: «KAP-Company» + “ACS for oven Acheson”		
III. Товар із підкріпленням	До продажу: <ul style="list-style-type: none">• можлива програма лояльності при підписаному довготривалому контракті;• Розрахунок готівкою/безготівковий;• пільги доставки.		
	Після продажу: <ul style="list-style-type: none">• гарантійний ремонт,• гарантія повернення грошей і обміну товар.		
За рахунок чого потенційний товар буде захищено від копіювання: буде розроблено патент на винахід, на кожному приладі буде акцизна марка та фірмовий штамп.			

Отже, переглянувши всі плюси та мінуси технології графітування електродів завдяки печі Ачесона, можна зробити висновки, що ідея є актуальною та нагальною для застосування. Програмне забезпечення та грамотно спроектована система автоматизованого керування дає можливість зменшити кількість обслуговуючого персоналу, підвищити якісь продукції, зменшити

витрати часу та надає можливість керувати процесом на відстані. Технологія є доступною в плані ціни та застосування на підприємствах чи установах.

ВИСНОВОК

В даній роботі розглядається процес виготовлення графітованої електродної продукції.

Для печі Ачесона була розроблена математична модель в програмному забезпеченні *ANSYS*, для отримання експериментальних даних та подальшого їх використання для створення спрощеної моделі процесу графітування печі Ачесона в математичному пакеті *MATLAB*.

Результати вказують на те, що спрощена модель вийшла порівняно точною та функціонує значно швидше ніж експериментальна модель.

Розроблено систему керування піччю графітування, проведено попередній аналіз існуючих систем керування. Під час розробки враховувались як теоретичні дані, так і експериментальні дані, що зіграло позитивну роль у якості функціонування всієї системи загалом.

Також, було створено прикладне програмне забезпечення мовою *C# .Net* для імітації, а також з можливістю керування у реальному часі, процесу графітування у печах Ачесона.

Останній розділ присвячений стартап проекту для даного процесу.

ЛІТЕРАТУРА

1. **Жученко О. А., Корж А. П.** Класифікація електродної продукції та її застосування [Текст] / А. І. Жученко, А. П. Корж – НТУУ "КПІ", 2016. – 90 с.
2. **Санников А. К.** Производство электродной продукции [Текст] / А. К. Санников, А. Б. Сомов, В. В. Ключников и др. – М.: Metallurgiya, 1985 г. – 129 с. – Библиогр.: с. 128. – 1230 экз.
3. **Шулепов С. В.** Физика углеграфитовых материалов [Текст] : монографія. – Ч.: ЧГПИ, 1968г. – 319 с.: іл. – Бібліогр.: 201–207. – 300 прим.
4. **Жученко А. І., Коржик М. В., Кутузов С. В.** Керування процесом графітації при виробництві електродної продукції [Текст] : монографія. – К.: НТУУ «КПІ», Вид. дім «Києво-Могилянська академія», 2013. – 224 с.: іл. – Бібліогр.: 201–207. – 300 прим.
5. **Розенберг В.Л.** Метод расчета индуктивностей токопроводящей системы «кернаграфитировочной печи – боковые шинопакеты» / В.Л. Розенберг, В.А. Сычев, Я.В. Метелица // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск: ЧПИ, 1985. – С. 15–18.
6. **Сычев В.А.** Расчет индуктивностей некоторых конструкций токопроводящей системы боковые шинопакеты – графитовочная печь / В.А. Сычев // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск : ЧПИ, 1986. – С. 23–25.
7. **Панов А.К.** Растекание тока в углеродистых образцах / А.К. Панов, А.А. Попова // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1973. – Вып. 5. – С. 101–105.
8. **Огнева М.Ф.** Механизм нагревания цилиндрической электродной заготовки в графитировочной печи / М.Ф. Огнева, В.П. Соседов, Е.Ф. Чалых, Н.И. Чичулин // Цветные металлы. – 1974. – № 10. – С. 42–45.
9. **Чичулин Н.И.** К вопросу графитации крупногабаритных электродов / Н.И. Чичулин, Е.И. Евсеев // Вопросы технического прогресса в электродной

промышленности : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1971. – Вып. 3. – С.162–169.

10. **Ахметшин Н.Ф.** Электрический и температурный режим графитации электродных заготовок / Н.Ф. Ахметшин, Е.Ф. Чалых, Е.Н. Шабуров [и др.] // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1975. – Вып. 7. – С. 44–51.

11. **Фридман А.М.** Методика моделирования тепловых полей печей графитации / А.М. Фридман, М.Г. Аветьян, Н.А. Михайлова [и др.] // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgia, 1978. – Вып. 13. – С. 6–11.

12. **Огнева М.Ф.** Степень влияния градиента температур в заготовка на оптимизацию режимов графитации / М.Ф. Огнева, А.Д. Кокурин, В.Ю. Смирнова [и др.] // Производство углеродных материалов : Сб. научн. тр. – М. : НИИГрафит, 1981. – С. 75–82.

13. **Короленко Ю.П.** К вопросу об электромоделировании нестационарных температурных полей / Ю.П. Короленко, П.П. Плотников // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск : ЧПИ, 1966. – С. 161–165.

14. **Знамеровский В.Ю.** Математическое моделирование процессов теплообмена в электрических печах сопротивления при производстве электродного графита / В.Ю. Знамеровский, В.В. Яшкина // Промышленная энергетика. – 1984. – № 2. – С. 31–33.

15. **Знамеровский В.Ю.** Особенности решения задач теплопроводности с внутренним источником теплоты / В.Ю. Знамеровский // Промышленная энергетика. – 1986. – № 3. – С. 24–26.

16. **Кузнецов Д.М., Фокин В.П.** Процесс графитации углеродных материалов. Современные методы исследования / Д.М. Кузнецов, В.П. Фокин. – Новочеркасск : ЮРГТУ, 2001. – 132 с.

17. **Горбенко В.И.** О выборе интервала времени при численном методе расчета температурных полей / В.И. Горбенко // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск : ЧПИ, 1973. – С. 113–116.

18. **Горбенко В.И.** Некоторые методы численного решения уравнения теплопроводности при переменных коэффициентах / В.И. Горбенко // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск : ЧПИ, 1975. – С. 172–177.

19. **Черненко И.И.** Расчет термонапряжений в трехслойном цилиндре из графитового материала с учетом специфических свойств внутреннего слоя / И.И. Черненко, В.И. Артемьев // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgy, 1972. – Вып. 7. – С. 146–157.

20. **Черненко И.И.** Расчет термических напряжений в цилиндре из анизотропного углеграфитового материала / И.И. Черненко, В.А. Кравцов, Г.В. Розанова // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgy, 1969. – Вып. 4. – С. 124–128.

21. **Николаев Н.Н.** Термическое напряжение в цилиндрических заготовках при графитации / Н.Н. Николаев, А.Г. Ганзен, В.А. Китова [и др.] // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgy, 1975. – Вып. 10. – С. 48–58.

22. **Горбенко В.И.** О генерации сеток при решении задач теплопроводности по методу балансов энергии / В.И. Горбенко // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск : ЧПИ, 1986. – С. 75–78.

23. **Короленко Ю.А.** К определению интервала времени при расчете сопряженных задач теплопроводности по неявной схеме / Ю.А. Короленко // Автоматизация энергосистем и энергоустановок промышленных предприятий : Сб. научн. тр. – Челябинск : ЧПИ, 1989. – С. 15–18.

24. **Доржиев М.Н.** Влияние параметров процесса графитации на качество

графитированных электродов / М.Н. Доржиев, Е.И. Евсеев, Е.Г. Шляпин [и др.] // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1975. – Вып. 7. – С. 13–23.

25. **Кузин Б.М.** Совершенствование технологии производства электродов диаметром 555 мм / Б.М. Кузин, Е.Н. Шабуров, А.А. Алексеев [и др.] // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1975. – Вып. 7. – С. 23–28.

26. **Павловский А.М.** Интенсификация процесса графитации электродов на Челябинском электродном заводе / А.М. Павловский, З.В. Столярова, Н.П. Подольская [и др.] // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1976. – Вып. 8. – С. 48–53.

27. **Миснар А.** Теплопроводность твердых тел, жидкостей, газов и их композиций / А. Миснар. – М. : Мир, 1968. – 460 с.

28. **Чичулин Н.И.** Зависимость электропроводности углеродистых материалов от температуры при графитации / Н.И. Чичулин, Е.Л. Петров // Вопросы технического прогресса в электродной промышленности : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1971. – Вып. 3. – С. 107–112.

29. **Ахметшин Н.Ф.** Влияние свойств междуэлектродной пересыпки на электрические, тепловые показатели и качество электродов / Н.Ф. Ахметшин, М.Н. Доржиев, Е.Н. Шабуров [и др.] // Вопросы технического прогресса в электродной промышленности : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1971. – Вып. 3. – С. 205–213.

30. **Чарыкова Л.Н.** Зависимость теплопроводности сыпучих углеродистых материалов различного грансостава от температуры и давления / Л.Н. Чарыкова, Н.И. Чичулин, Е.И. Евсеев // Техника и технология графитированных электродов : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1972. – Вып. 4. – С. 47–53.

31. **Горпиненко М.С.** Современное состояние и научно обоснованные требования к нефтяному коксу – сырью для производства графитированных

электродов / М.С. Горпиненко, В.В. Зеленина, И.Ф. Сухоруков [и др.] // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1975. – Вып. 7. – С. 102–112.

32. **Лутков А.И.** Исследование теплофизических свойств углеграфитового материала в зависимости от температуры и времени термической обработки / А.И. Лутков, С.Е. Вяткин, Б.К. Дымов [и др.] // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgia, 1966. – Вып. 2. – С. 88–98.

33. **Аверина М.В.** Зависимость свойств коксов от их структуры / М.В. Аверина, А.И. Николаев, Л.З. Синельников // Цветные металлы. – 1975. – № 5. – С. 43–47.

34. **Гуревич Л.В.** Термодинамические свойства индивидуальных веществ: Справочное издание в 4-х т. / Л.В. Гуревич, И.В. Вейц, В.А. Медведев [и др.] – Т. 2 ; Кн. 1. – М. : Наука, 1979. – 440 с.

35. **Лутков А.И.** Тепловые и электрические свойства углеродных материалов / А.И. Лутков. – М. : Metallurgia, 1990. – 196 с.

36. **Лутков А.И.** Тепло- и электропроводность конструкционного графита в интервале 80–2500°K / А.И. Лутков, Б.К. Дымов, В.И. Волга // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgia, 1969. – Вып. 4. – С. 59–66.

37. **Виноградова К.П.** Исследование свойств графита на основе крекингowego кокса / К.П. Виноградова, Н.О. Гусман, В.А. Черных [и др.] // Конструкционные материалы на основе графита : Сб. научн. тр. – М. : Metallurgia, 1981. – Вып. 16. – С. 67–71.

38. **Чичулин Н.И.** Теплоизолирующие материалы графитировочных печей / Н.И. Чичулин, М.Ф. Огнева, О.В. Мокрушина [и др.] // Совершенствование технологии и улучшение качества электродной продукции : Сб. научн. тр. – Челябинск : ГосНИИЭП, 1975. – Вып. 7. – С. 176–182.

39. **Лутков А.И.** Тепло- и электропроводность пересыпки при графитации / А.И. Лутков, В.Н. Михайлов, Б.К. Дымов // Цветные металлы. – 1975. – № 5. – С. 41–43.

40. Разработка математической модели промышленного процесса графитации с целью его оптимизации и усовершенствования : Отчет о НИР ГосНИИЭП, Инв. № 02850062203 / [руководитель В.Ю. Знамеровский] ; опубл в сб. реф. НИР и ОКР, ВНТИЦентр. – Серия 17. – 1986. – № 15. – Реф. 61.15.86.165.

41. **Кузнецов Д.М., Фокин В.П.** Процесс графитации углеродных материалов. Современные методы исследования / Д.М. Кузнецов, В.П. Фокин. – Новочеркасск : ЮРГТУ, 2001. – 132 с.

42. **Знамеровский В.Ю.** Методика исследования промышленных печей графитации / В.Ю. Знамеровский, Д.М. Кузнецов, В.А. Коцюр [и др.] // Промышленная энергетика. – 1988. – № 9. – С. 32–34.

43. **Чалых А.Ф.** Графитация углеродистых материалов [Текст] / А.Ф. Чалых, В.П. Соседов – М.:Металлургия, 1987 г. – 176 с. – Библиогр.: с. 176. – 1230 экз.

44. **Жученко А. І.** Розробка спрощеної математичної моделі скловарної печі [текст] / А. І. Жученко, В.С. Цапар // Східно-Європейський журнал передових технологій, №2/4 (74), 2015. – с. 42-48

45. **Зайцев В. Ф., Полянин А. Д.** Метод разделения переменных в математической физике [Текст] / В.Ф. Зайцев, А.Д. Полянин – СПб., 2009. – 92 с. – ISBN 978–5–94777–211–1. (СК_)

ДОДАТКИ

Додаток 1 – Лістинг програмного забезпечення програми мовою *C#.Net*

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Drawing.Design;
using Microsoft.Office.Interop.Excel;
using Excel = Microsoft.Office.Interop.Excel;
using Application = Microsoft.Office.Interop.Excel.Application;
using System.Diagnostics;

namespace Pech1
{
    using MyCharts =
System.Windows.Forms.DataVisualization.Charting.Chart;
    public partial class Form1 : Form
    {
        List<ChartValues> _LstChartValues = null;
        List<MyCharts> _LstCharts = null;
        Timer t, t1, t2, t3, t4, t5 = null;
        int img = 0;
        bool direc = false;
        bool CV = false;
        string _path = String.Empty;
        int ticksCout = 0;
        int ticksMoney = 0;
        double ampCount = 0;
        int i = 0;
        int N;
        float amper = 0;
        float volt = 0;
        double power = 0;
        double cost = 0;
        double cost1 = 0;
        double howmuch = 0;
        Random r = new Random();
        double v1 = 0;
        double v2 = 0;
        double v3 = 0;
        double T1 = 0;
        double T2 = 0;
        double T3 = 0;
        double Perepad = 0;
        double Perepad1 = 0;
        double Perepad2 = 0;
        double rnd = 0;
        double rnd1 = 0;
        double rnd2 = 0;
        double rnd3 = 0;
        double rnd31 = 0;
        double rnd32 = 0;
        double rnd4 = 0;
        double rnd5 = 0;
        double rnd6 = 0;
        double volt2 = 0;
        double power2 = 0;
        double energy = 0;
        double energy1 = 0;
    }
}

```



```

Excel.Application xlApp = new Excel.Application();

public Form1()
{
    InitializeComponent();
    InitTimer();
    _LstChartValues = new List<ChartValues>();
    LoadChartsToList();
}

private void LoadChartsToList()
{
    _LstCharts = new List<MyCharts>();

    _LstCharts.Add(chart1);
    _LstCharts.Add(chart2);
    _LstCharts.Add(chart3);
    _LstCharts.Add(chart4);
    _LstCharts.Add(chart5);
    _LstCharts.Add(chart6);
    _LstCharts.Add(chart7);
    _LstCharts.Add(chart8);
    _LstCharts.Add(chart9);
    _LstCharts.Add(chart10);
    _LstCharts.Add(chart11);
}

private void ClearCharts()
{
    v1 = 0;
    v2 = 0;
    v3 = 0;
    Perepad = 0;
    Perepad1 = 0;
    Perepad2 = 0;
    T1 = 0;
    T2 = 0;
    T3 = 0;
    ampCount = 0;
    power2 = 0;
    cost = 0;
    power = 0;
    cost1 = 0;
    N = 0;
    _LstChartValues.Clear();
    chart1.Series[0].Points.Clear();
    chart2.Series[0].Points.Clear();
    chart3.Series[0].Points.Clear();
    chart4.Series[0].Points.Clear();
    chart5.Series[0].Points.Clear();
    chart6.Series[0].Points.Clear();
    chart7.Series[0].Points.Clear();
    chart8.Series[0].Points.Clear();
    chart9.Series[0].Points.Clear();
    chart10.Series[0].Points.Clear();
    chart11.Series[0].Points.Clear();
}

public void Form_Load(object sender, EventArgs e)
{
}

public void InitTimer()
{
    t = new Timer();
    t.Interval = 400;
}

```

```

t.Tick += T_Tick;

t1 = new Timer();
t1.Interval = 500;
t1.Tick += T1_Tick;

t2 = new Timer();
t2.Interval = 100;
t2.Tick += T2_Tick;

t3 = new Timer();
t3.Interval = 500;
t3.Tick += T3_Tick;

t4 = new Timer();

t4.Tick += T4_Tick;

t5 = new Timer();
t5.Interval = 200;
t5.Tick += T5_Tick;

}

private void ChartVisible (bool CV)
{
    if (CV==true)
    {
        chart1.Visible = true;
        chart2.Visible = true;
        chart3.Visible = true;
        chart4.Visible = true;
        chart5.Visible = true;
        chart6.Visible = true;
        chart7.Visible = true;
        chart8.Visible = true;
        chart9.Visible = true;
        chart10.Visible = true;
        chart11.Visible = true;
    }
    else
    {
        chart1.Visible = false;
        chart2.Visible = false;
        chart3.Visible = false;
        chart4.Visible = false;
        chart5.Visible = false;
        chart6.Visible = false;
        chart7.Visible = false;
        chart8.Visible = false;
        chart9.Visible = false;
        chart10.Visible = false;
        chart11.Visible = false;
    }
}

private void DrawChart(int N)
{
    for (int i = 0; i < 11; i++)
    {
        if (i < 3)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0, N);
            _LstCharts[i].Series[0].Points.AddXY(N, GetV(i));
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;

```

```

        _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
        _LstChartValues.Add(new ChartValues((i + 1).ToString(),
N.ToString(), GetV(i).ToString()));
    }
    else if (i >= 3 & i < 6)
    {
        _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0, N);
        _LstCharts[i].Series[0].Points.AddXY(N, GetT(i));
        _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
        _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
        _LstChartValues.Add(new ChartValues((i + 1).ToString(),
N.ToString(), GetT(i).ToString()));
    }
    else
    {
        if (i == 6)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);

            _LstCharts[i].Series[0].Points.AddXY(N, Perepad);
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), Perepad.ToString()));
        }
        if (i == 7)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);

            _LstCharts[i].Series[0].Points.AddXY(N, volt2);
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), volt2.ToString()));
        }
        if (i == 8)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);

            _LstCharts[i].Series[0].Points.AddXY(N, ampCount);
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), ampCount.ToString()));
        }
        if (i == 9)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);

            _LstCharts[i].Series[0].Points.AddXY(N, Perepad1);
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), Perepad1.ToString()));
        }
        if (i == 10)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);

            _LstCharts[i].Series[0].Points.AddXY(N, Perepad2);
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), Perepad2.ToString()));
        }
    }
}
}

```

```

    }

    private double GetT(int i)
    {
        double retVal = 0;
        switch (i)
        {
            case 3:
                retVal = T1;
                break;
            case 4:
                retVal = T2;
                break;
            case 5:
                retVal = T3;
                break;
        }
        return retVal;
    }

    private double GetV(int i)
    {
        double retVal = 0;
        switch (i)
        {
            case 0:
                retVal = v1;
                break;
            case 1:
                retVal = v2;
                break;
            case 2:
                retVal = v3;
                break;
        }
        return retVal;
    }

    public void T5_Tick(object sender, EventArgs e)
    {
        pictureBox1.Visible = false;
        GreatPB.Visible = true;
        GreatPB.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\Pich\1\peopl1.png");
        GreatPB.Location = new
System.Drawing.Point(GreatPB.Location.X - 10, GreatPB.Location.Y);
        if (GreatPB.Location.X <= 250) // и здесь нужно подобрать
правильно отступы
        { GreatPB.Visible = false;
          t5.Stop();
          pictureBox1.Visible = true;
          pictureBox1.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\Pich\1\Avariya1.png");
          GreatPB.Location = new System.Drawing.Point(516,
GreatPB.Location.Y);
        }
        //MoneyPB.Location.X >= Width - MoneyPB.Width - 220
    }

    public void T4_Tick(object sender, EventArgs e)
    {
        N++;
        DrawChart(N);
        progressBar1.Increment(1);
        if (progressBar1.Value == 220)
        {
            progressBar1.Value = 0;

```

```

    }
    volt2 = r.Next(180, 230);
    VoltageCL.Text = (volt2).ToString();
    AmperCL.Text = Math.Round((ampCount += 0.2), 3).ToString();
    CurrentTB.Text = Math.Round((ampCount += 0.2), 3).ToString();
    VoltageTB.Text = Math.Round((volt2), 3).ToString();

    if (ampCount >= 88)
    {
        progressBar1.Value = 220;
        StopAllTier();
        IndicatorGrafCompleat.Text = "Вітаємо. Графітування завершено";
        IndicatorLB.Text = "Вітаємо. Графітування завершено";
        MoneyPB.Location = new System.Drawing.Point(737,
MoneyPB.Location.Y);
        GreatPB.Visible = true;
        GreatPB.Image = Image.FromFile(Directory.GetCurrentDirectory()
+ @"\Pich\1\great.png");
        SSEkrPB.Image = Image.FromFile(Directory.GetCurrentDirectory()
+ @"\Pich\1\stop.png");
    }
    var n = 220;
    var a = new double[n];
    var b = new double[n];
    var c = new double[n];
    var d = new double[n];
    var d2 = new double[n];
    var d1 = new double[n];
    var f = new double[n];
    var g = new double[n];
    var h = new double[n];
    for (var i = 0; i < n; i++)
    a[i] = r.NextDouble(-0.13, 1.24);
    b[i] = r.NextDouble(0.43, 1.34);
    c[i] = r.NextDouble(0.23, 1.14);
    d[i] = r.NextDouble(4.15, 5.15);
    d1[i] = r.NextDouble(4.52, 5.55);
    d2[i] = r.NextDouble(4.5, 5.45);
    f[i] = r.NextDouble(-0.95, 3.05);
    g[i] = r.NextDouble(3.01, 3.06);
    h[i] = r.NextDouble(3.09, 4.07);

    rnd = a[i];
    rnd1 = b[i];
    rnd2 = c[i];
    rnd3 = d[i];
    rnd31 = d1[i];
    rnd32 = d2[i];
    rnd4 = f[i];
    rnd5 = g[i];
    rnd6 = h[i];

    v1 += Math.Round(rnd, 3);
    v2 += Math.Round(rnd1, 3);
    v3 += Math.Round(rnd2, 3);
    Perepad += Math.Round(rnd3, 3);
    Perepad1 += Math.Round(rnd31, 3);
    Perepad2 += Math.Round(rnd32, 3);
    T1 += Math.Round(rnd4, 3);
    T2 += Math.Round(rnd5, 3);
    T3 += Math.Round(rnd6, 3);
    howmuch = Convert.ToDouble(howmuchTB.Text);

    power2 = Math.Round((ampCount * volt2), 3);
    energy = Math.Round((power2 / 3600), 3);
    cost = energy * howmuch;
    MoneyLB.Text = (cost1 += cost).ToString();

```

```

        EnergyCL.Text = (energy1 += energy).ToString();
        PowerCL.Text = power2.ToString();
        PowerTB.Text = power2.ToString();
    }

    public void T3_Tick(object sender, EventArgs e)
    {
        if (i == 0)
        {
            StrON.Location = new
System.Drawing.Point(StrON.Location.X + 10, StrON.Location.Y);
            if (StrON.Location.X >= Width - StrON.Width - 850)// и
здесь нужно подобрать правильно отступы
                i++;
            return;
        }
        if (i == 1)
        {
            StrON.Location = new System.Drawing.Point(485,
StrON.Location.Y);
            i = 0;
            return;
        }
    }

    void T2_Tick(object sender, EventArgs e)
    {
        if (i == 0)
        {
            ticksMoney++;
            MoneyPB.Location = new
System.Drawing.Point(MoneyPB.Location.X + 10, MoneyPB.Location.Y);
            if (MoneyPB.Location.X >= Width - MoneyPB.Width -
150)// и здесь нужно подобрать правильно отступы
                i++;

            return;
        }
        if (i == 1)
        {
            MoneyPB.Location = new
System.Drawing.Point(MoneyPB.Location.X - 10, MoneyPB.Location.Y);
            if (MoneyPB.Location.X <= 737) //здесь
                i = 0;
            return;
        }
    }

    public void T1_Tick(object sender, EventArgs e)
    {
        if (img <= 4 & !direc)
        {
            _path = Directory.GetCurrentDirectory() +
$@"\Pich\2\{img}.png";
            img++;
            pictureBox1.Image = Image.FromFile(_path);

            if (img == 3)
            {
                direc = true;
            }
        }
        if (img > 0 & direc)
        {

```

```

    _path = Directory.GetCurrentDirectory() +
    $"{@"\Pich\2\{img}.png"}";
    img--;
    pictureBox1.Image = Image.FromFile(_path);

    if (img == 0)
        direc = false;
    }
}

private void T_Tick(object sender, EventArgs e)
{
    ticksCout++;
    if (img <= 7 & !direc)
    {
        amper = r.Next(2, 6);
        CurrentTB.Text = amper.ToString();
        volt = r.Next(219, 222);
        VoltageTB.Text = volt.ToString();
        power = (amper * volt);
        PowerTB.Text = (power).ToString();

        _path = Directory.GetCurrentDirectory() + $"{@"\Pich\{img}.png"}";
        img++;
        pictureBox1.Image = Image.FromFile(_path);

        if (img == 6)
        {
            direc = true;
        }
    }
    if (img > 0 & direc)
    {
        amper = r.Next(1, 4);
        CurrentTB.Text = amper.ToString();
        volt = r.Next(219, 222);
        VoltageTB.Text = volt.ToString();
        power = (amper * volt);
        PowerTB.Text = (power).ToString();

        _path = Directory.GetCurrentDirectory() + $"{@"\Pich\{img}.png"}";
        img--;
        pictureBox1.Image = Image.FromFile(_path);

        if (img == 0)
            direc = false;
    }

    if (ticksCout == 20)
    {
        t3.Start();
        StrON.Visible = true;
        IndicatorLB.Text = "Індикатор стану печі. Наразі: Піч готова
до процесу";
        IndicatorGrafCompleet.Text = "Індикатор стану печі. Наразі: Піч
готова до процесу";

        t.Stop();
        ticksCout = 0;
        pictureBox1.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"Pich\1\8.png");
        StartGrafit.Visible = true;
        power = (amper * volt2);
        PowerTB.Text = (power).ToString();
        SSEkrPB.Image = Image.FromFile(Directory.GetCurrentDirectory()
+ @"Pich\1\pause.png");
    }
}

```

```

        StrON.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\"Pich\1\arrow.png");
        EkranPB.Visible = false;
        EkranPB_click.Visible = true;
    }

}

private void StartButton_Click(object sender, EventArgs e)
{
    t.Start();
    GreatPB.Visible = false;
    StopButton.Visible = true;
    StartButton.Visible = false;
    IndicatorLB.Text = "Індикатор стану печі. Наразі: Піч
проходить підготовку";
    IndicatorGrafCompleet.Text = "Індикатор стану печі. Наразі:
Піч проходить підготовку";

    CurrentTB.Visible = true;
    VoltageTB.Visible = true;
    PowerTB.Visible = true;
    MoneyPB.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\"Pich\1\dolar1.png");
    progressBar1.Value = 0;
    SSEkrPB.Visible = true;
    SSEkrPB.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\"Pich\1\start1.png");
    RegimeTrackBar.Value = 2;
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void StartGrafit_Click(object sender, EventArgs e)
{
    t4.Start();
    t2.Start();
    t1.Start();
    MoneyPB.Visible = true;
    StartGrafit.Visible = false;
    IndicatorLB.Text = "Індикатор стану печі. Наразі: Піч
графітації працює у робочому режимі";
    IndicatorGrafCompleet.Text = "Індикатор стану печі. Наразі:
Піч графітації працює у робочому режимі";
    GreatPB.Visible = false;
    Cleaner(xlApp, false);
    SSEkrPB.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\"Pich\1\start1.png");
    ChartVisible(CV = true);
}

private void Stop_button2_Click(object sender, EventArgs e)
{
    StopAllTier();
    ampCount = 0;
    pictureBox1.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\"Pich\0.png");
    StartButton.Visible = true;
    StopButton.Visible = false;
    IndicatorLB.Text = "Індикатор стану печі. Наразі: Піч
зупинена :(";
    IndicatorGrafCompleet.Text = "Індикатор стану печі. Наразі:
Піч зупинена :(";
    StartGrafit.Visible = false;

```



```

        MoneyPB.Visible = false;
        StrON.Visible = false;
        EkranPB_click.Visible = true;
        EkranPB.Visible = true;
        GreatPB.Visible = false;
        StartGrafit.Visible = true;
        SSEkrPB.Image = Image.FromFile(Directory.GetCurrentDirectory() +
@"\Pich\1\stop.png");
        RegimeTrackBar.Value = 2;
        ClearCharts();
        ChartVisible(CV = false);
    }

    private void EkranPB_click_Click(object sender, EventArgs e)
    {
        EkranPB_click.Visible = false;
        t3.Stop();
        StrON.Visible = false;
        MoneyPB.Visible = true;
    }

    private void howmuchTB_KeyPress(object sender, KeyPressEventArgs e)
    {
        if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
        {
            return;
        }
        if (e.KeyChar == '.')
        {
            e.KeyChar = ',';
        }
        if (e.KeyChar == ',')
        {
            if (howmuchTB.Text.IndexOf(',') != -1)
            {
                e.Handled = true;
            }
            return;
        }
        if (Char.IsControl(e.KeyChar))
        {
            if (e.KeyChar == (char)Keys.Enter)
                StartGrafit.Focus();
            return;
        }
        e.Handled = true;

        //if (!Char.IsDigit(e.KeyChar) && e.KeyChar != 8)
        //    e.Handled = true;
    }

    private void Cleaner(Excel.Application xlApp, bool KillProcess)
    {
        try
        {
            //Excel object
            if (xlApp != null)
            {
                xlApp.Quit();
                xlApp = null;
            }
        }
        catch (Exception)
        {
        }
    }

```

```

//Excel Process
if (KillProcess)
{
    Process[] prc = Process.GetProcesses();
    for (byte i = 0; i < prc.Length; i++)
        if (prc[i].ProcessName.Equals("EXCEL"))
            prc[i].Kill();
}

//clear temp links and junk
System.GC.Collect();
}

private void buttonToExcel_Click(object sender, EventArgs e)
{
    string fileName = Directory.GetCurrentDirectory() +
@"\Pich\1\ExcelFile.xlsx"; //@"C:\Pich\1\ExcelFile.xlsx"; //
    Excel.Workbook xlWb = xlApp.Workbooks.Open(fileName);
    //відкриваємо Excel файл
    Excel.Worksheet xlSht = xlWb.Sheets[1]; //перший лист в
    файлі

    int iLastRow = xlSht.Cells[xlSht.Rows.Count,
    "A"].End[Excel.XlDirection.xlUp].Row; //остання заповнена строка в стовбці A
    Excel.Range xRange = xlSht.Range["A1", "N300"];
    xRange.NumberFormat = "@";

    xlSht.Cells[1, "A"].Value = "№пп";
    xlSht.Cells[1, "B"].Value = "v1";
    xlSht.Cells[1, "C"].Value = "v2";
    xlSht.Cells[1, "D"].Value = "v3";
    xlSht.Cells[1, "E"].Value = "t1";
    xlSht.Cells[1, "F"].Value = "t2";
    xlSht.Cells[1, "G"].Value = "t3";
    xlSht.Cells[1, "H"].Value = "Перепад1";
    xlSht.Cells[1, "I"].Value = "Напруга";
    xlSht.Cells[1, "J"].Value = "Сила струму";
    xlSht.Cells[1, "K"].Value = "Перепад2";
    xlSht.Cells[1, "L"].Value = "Перепад3";
    xlSht.Cells[1, "M"].Value = "Час запису";

    bool f = false;
    if (!f)
    { iLastRow++; f = true; }

    for (int i = 0; i < _LstChartValues.Count; i++)
    {
        System.Diagnostics.Debug.WriteLine($"i =
{i.ToString()} :: { _LstChartValues[i].ChartId} -
{ _LstChartValues[i].ValueY}");
    }
    for (int k = 0; k < _LstChartValues.Count; k++)
    {
        xlSht.Cells[iLastRow, "M"].Value = DateTime.Now; ;
        xlSht.Cells[iLastRow, "A"].Value = (iLastRow -
1).ToString();

        switch ( _LstChartValues[k].ChartId)
        {
            case "1":
                xlSht.Cells[iLastRow, "B"].Value =
_LstChartValues[k].ValueY;
                break;
            case "2":
                xlSht.Cells[iLastRow, "C"].Value =
_LstChartValues[k].ValueY;
                break;

```

```

        case "3":
            xlSht.Cells[iLastRow, "D"].Value =
_LstChartValues[k].ValueY;
            break;
        case "4":
            xlSht.Cells[iLastRow, "E"].Value =
_LstChartValues[k].ValueY;
            break;
        case "5":
            xlSht.Cells[iLastRow, "F"].Value =
_LstChartValues[k].ValueY;
            break;
        case "6":
            xlSht.Cells[iLastRow, "G"].Value =
_LstChartValues[k].ValueY;
            break;
        case "7":
            xlSht.Cells[iLastRow, "H"].Value =
_LstChartValues[k].ValueY;
            break;
        case "8":
            xlSht.Cells[iLastRow, "I"].Value =
_LstChartValues[k].ValueY;
            break;
        case "9":
            xlSht.Cells[iLastRow, "J"].Value =
_LstChartValues[k].ValueY;
            break;
        case "10":
            xlSht.Cells[iLastRow, "K"].Value =
_LstChartValues[k].ValueY;
            break;
        case "11":
            xlSht.Cells[iLastRow, "L"].Value =
_LstChartValues[k].ValueY;
            iLastRow++;
            break;
    }
}
xlWb.Close(true); //закрити і зберегти книгу
xlApp.Quit();
MessageBox.Show("Файл успішно збережено!");
}
private void StopAllTier()
{
    t.Stop();
    t1.Stop();
    t2.Stop();
    t3.Stop();
    t4.Stop();
    t5.Stop();
}

private void RegimTrackBar_Scroll(object sender, EventArgs e)
{
    if (RegimeTrackBar.Value == 1)
    {
        StopAllTier();
        ampCount = 0;
        StartButton.Visible = false;
        IndicatorLB.Text = "Індикатор стану печі. Наразі: Пич
зупинена. Аварія";
        IndicatorGrafCompleat.Text = "Індикатор стану печі. Наразі: Пич
зупинена. Аварія";
        StartGrafit.Visible = false;
        MoneyPB.Visible = false;
    }
}

```

```

        StrON.Visible = false;
        GreatPB.Visible = false;
        SSEkrPB.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\Pich\1\stop.png");
        t5.Start();
    }
    else
    {
        IndicatorLB.Text = "Натисніть кнопку STOP для
налаштування";
        IndicatorGrafCompleat.Text = "Натисніть кнопку STOP
для налаштування";
    }
}

private void StopButton_Click(object sender, EventArgs e)
{
    StopAllTier();
    ampCount = 0;
    pictureBox1.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\Pich\0.png");
    StartButton.Visible = true;
    StopButton.Visible = false;
    IndicatorLB.Text = "Індикатор стану печі. Наразі: Печ
зупинена :(";
    IndicatorGrafCompleat.Text = "Індикатор стану печі. Наразі:
Печ зупинена :(";
    StartGrafit.Visible = false;
    MoneyPB.Visible = false;
    StrON.Visible = false;
    EkranPB_click.Visible = true;
    EkranPB.Visible = true;
    GreatPB.Visible = false;
    SSEkrPB.Image =
Image.FromFile(Directory.GetCurrentDirectory() + @"\Pich\1\stop.png");
    RegimeTrackBar.Value = 2;
    ClearCharts();
    ChartVisible(CV = false);
}
}
}

```

Додаткові класи:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Pech1
{
    public class ChartValues
    {
        private string _ValueX = String.Empty;
        private string _ValueY = String.Empty;
        private string _ChartId = String.Empty;

        public string ValueX
        {
            get { return this._ValueX; }
            set { this._ValueY = value; }
        }

        public string ValueY
        {
            get { return _ValueY; }
        }
    }
}

```

```

        set { _ValueY = value; }
    }

    public string ChartId
    {
        get { return _ChartId; }
        set { _ChartId = value; }
    }

    public ChartValues(string X, string Y)
    {
        _ValueX = X;
        _ValueY = Y;
    }
    public ChartValues(string ChartId, string X, string Y)
    {
        _ValueX = X;
        _ValueY = Y;
        _ChartId = ChartId;
    }
    }
}

//-----

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Pech1
{
    public static class DRand
    {
        public static double NextDouble(this Random random, double minValue,
double maxValue)
        {
            return random.NextDouble() * (maxValue - minValue) + minValue;
        }
    }
}

//-----

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Pech1
{
    class TimerGraf
    {
        TimerGraf TG = new TimerGraf();
        private void StartGrafit2_Click(object sender, EventArgs e)
        {

        }

        public static void TimerGraf_Stop()
        {

        }
    }
}

//-----

```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows.Forms;

namespace Pech1
{
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Додаток 2 – Лістинг програмного забезпечення програми мовою *WPF .Net*

```
<Application x:Class="PichWPF.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:PichWPF"
    StartupUri="View.xaml">
    <Application.Resources>

    </Application.Resources>
</Application>
```

```
//-----
```

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;
using System.Windows;

namespace PichWPF
{
    /// <summary>
    /// Interaction logic for App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}
```

```
//-----
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PichWPF
{
    public class ChartValues
    {
        private string _ValueX = String.Empty;
        private string _ValueY = String.Empty;
        private string _ChartId = String.Empty;

        public string ValueX
        {
            get { return this._ValueX; }
            set { this._ValueY = value; }
        }

        public string ValueY
        {
            get { return _ValueY; }
            set { _ValueY = value; }
        }

        public string ChartId
        {
            get { return _ChartId; }
            set { _ChartId = value; }
        }
    }
}
```

```

    public ChartValues(string X, string Y)
    {
        _ValueX = X;
        _ValueY = Y;
    }
    public ChartValues(string ChartId, string X, string Y)
    {
        _ValueX = X;
        _ValueY = Y;
        _ChartId = ChartId;
    }
}

//-----

<Window
    xmlns:wf="clr-namespace:System.Windows.Forms;assembly=System.Windows.Forms"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:PichWPF"
    xmlns:OPCWPFDashboard="clr-
namespace:OPCWPFDashboard;assembly=OPCWPFDashboard" x:Class="PichWPF.MainWindow"
    mc:Ignorable="d"
    Title="MainWindow" Height="700" Width="1000">
    <Window.Resources>

        <Style TargetType="{x:Type Label}">
            <Setter Property="Margin" Value="0,5,0 , 0"/>
        </Style>
    </Window.Resources>
    <StackPanel>

        <Grid>
            <Button x:Name="StartButton" Content="Start" HorizontalAlignment="Center"
Margin="0,20,0,0" VerticalAlignment="Top" Width="60" Height="30"
AutomationProperties.Name="StartButton" Click="StartButton_Click"/>
            <Button x:Name="StopButton" Content="Stop" HorizontalAlignment="Center"
Margin="0,60,0,0" VerticalAlignment="Top" Width="60" Height="30"
AutomationProperties.Name="StopButton" Click="StopButton_Click"/>
            <Image HorizontalAlignment="Right" Height="280" Margin="0,0,0,0"
VerticalAlignment="Top" Width="450" Source="ekran.png "
AutomationProperties.Name="ekran1IMG"/>
            <Image HorizontalAlignment="Left" Height="280" Margin="0,0,0,0"
VerticalAlignment="Top" Width="450" Source="ekran.png "
AutomationProperties.Name="ekran2IMG"/>
            <Image HorizontalAlignment="Center" Height="400" Margin="0,280,0,0"
VerticalAlignment="Top" Width="650" Source="ekran007.png "
AutomationProperties.Name="ekran3IMG"/>
            <Label x:Name="IndicatorLB" Content="Індикатор стану печі. Наразі: Піч
вимкнена" HorizontalAlignment="Left" Margin="20,10,0,0" VerticalAlignment="Top"
AutomationProperties.Name="IndicatorLB"/>
            <Label x:Name="CurrentLB1" Content="Сила струму, кА"
HorizontalAlignment="Left" Margin="20,30,0,0" VerticalAlignment="Top"
AutomationProperties.Name="CurrentLB1"/>
            <Label x:Name="VoltageLB1" Content="Напруга, В"
HorizontalAlignment="Left" Margin="140,30,0,0" VerticalAlignment="Top"
AutomationProperties.Name="VoltageLB1"/>
            <Label x:Name="PowerLB1" Content="Потужність, кВт"
HorizontalAlignment="Left" Margin="220,30,0,0" VerticalAlignment="Top"
AutomationProperties.Name="PowerLB1"/>
            <Label x:Name="CurrentTB" Content="Сила струму, кА"
HorizontalAlignment="Left" Margin="20,45,0,0" VerticalAlignment="Top"
AutomationProperties.Name="CurrentTB"/>

```



```

        <Label x:Name="VoltageTB" Content="Напруга, В" HorizontalAlignment="Left"
Margin="140,45,0,0" VerticalAlignment="Top" AutomationProperties.Name="VoltageTB"/>
        <Label x:Name="PowerTB" Content="Потужність, кВт"
HorizontalAlignment="Left" Margin="220,45,0,0" VerticalAlignment="Top"
AutomationProperties.Name="PowerTB"/>
        <Image x:Name="MoneyPB" HorizontalAlignment="Right" Height="35"
Margin="0,35,235,0" VerticalAlignment="Top" Width="35"
AutomationProperties.Name="MoneyPB" Source="dolar1.png" />
        <Image x:Name="SSEkrPB" HorizontalAlignment="Center" Height="32"
Margin="0,100,0,0" VerticalAlignment="Top" Width="32"
AutomationProperties.Name="SSEkrPB" />
        <Image x:Name="pictureBox1" HorizontalAlignment="Left" Height="169"
Margin="20,76,0,0" VerticalAlignment="Top" Width="324"
AutomationProperties.Name="pictureBox1" Source="8.png" />
        <Image x:Name="GreatPB" HorizontalAlignment="Left" Height="130"
Margin="349,115,0,0" VerticalAlignment="Top" Width="85"
AutomationProperties.Name="GreatPB" Source="great.png" />
        <Image x:Name="StrON" HorizontalAlignment="Center" Height="50" Margin="-
30,185,0,0" VerticalAlignment="Top" Width="50" AutomationProperties.Name="StrON"
Source="arrow.png" />
        <Label Content="Ціна на електроенергію, грн" HorizontalAlignment="Right"
Margin="0,30,270,0" VerticalAlignment="Top"
AutomationProperties.Name="howmuchTBName"/>
        <TextBox x:Name="howmuchTB" HorizontalAlignment="Right" Height="20"
Margin="0,50,350,0" TextWrapping="Wrap" Text="2,18" VerticalAlignment="Top"
Width="50" AutomationProperties.Name="howmuchTB"/>
        <Label Content="Використана енергія, кВт*год"
HorizontalAlignment="Right" Margin="0,65,265,0" VerticalAlignment="Top"
AutomationProperties.Name="EnergyCLName"/>
        <Label x:Name="EnergyCL" Content="EnergyCL" HorizontalAlignment="Right"
Margin="0,85,300,0" VerticalAlignment="Top" AutomationProperties.Name="EnergyCL"/>
        <Label Content="Потужність, кВт" HorizontalAlignment="Right"
Margin="0,105,300,0" VerticalAlignment="Top"
AutomationProperties.Name="PowerCLName"/>
        <Label x:Name="PowerCL" Content="PowerCL" HorizontalAlignment="Right"
Margin="0,122,315,0" VerticalAlignment="Top" AutomationProperties.Name="PowerCL"/>
        <Label Content="Сила струму, кА" HorizontalAlignment="Right"
Margin="0,140,340,0" VerticalAlignment="Top"
AutomationProperties.Name="AmperCLName"/>
        <Label x:Name="AmperCL" Content="4" HorizontalAlignment="Right"
Margin="0,160,380,0" VerticalAlignment="Top" AutomationProperties.Name="AmperCL"/>
        <Label Content="Напруга" HorizontalAlignment="Right"
Margin="0,140,275,0" VerticalAlignment="Top"
AutomationProperties.Name="VoltageCLName"/>
        <Label x:Name="VoltageCL" Content="220" HorizontalAlignment="Right"
Margin="0,160,290,0" VerticalAlignment="Top" AutomationProperties.Name="VoltageCL"/>
        <Button x:Name="StartGrafit" Content="StartGrafit"
HorizontalAlignment="Right" Margin="0,90,100,0" VerticalAlignment="Top" Width="60"
Height="30" AutomationProperties.Name="StartGrafit" Click="StartGrafit_Click"/>
        <Button x:Name="Stop_button2" Content="Stop" HorizontalAlignment="Right"
Margin="0,90,20,0" VerticalAlignment="Top" Width="60" Height="30"
AutomationProperties.Name="Stop_button2" Click="Stop_button2_Click"/>
        <Label x:Name="IndicatorGrafComple" Content="Індикатор стану печі.
Наразі: Печ вимкнена" HorizontalAlignment="Right" Margin="10,10,15,0"
VerticalAlignment="Top" AutomationProperties.Name="IndicatorGrafComple"/>
        <ProgressBar x:Name="progressBar1" HorizontalAlignment="Right"
Height="30" Margin="0,215,30,0" VerticalAlignment="Top" Width="390"
AutomationProperties.Name="progressBar1"/>
        <Label Content="Кошти, що витрачено" HorizontalAlignment="Right"
Margin="0,30,15,0" VerticalAlignment="Top" AutomationProperties.Name="MoneyTBName1"/>
        <Label Content="на електроенергію, грн" HorizontalAlignment="Right"
Margin="0,45,15,0" VerticalAlignment="Top" AutomationProperties.Name="MoneyTBName2"/>
        <Label x:Name="MoneyLB" Content="0,00" HorizontalAlignment="Right"
Margin="0,60,20,0" VerticalAlignment="Top" AutomationProperties.Name="MoneyLB"/>
        <Button x:Name="Recovery_Button" Content="Відновлення"
HorizontalAlignment="Right" Margin="0,160,50,0" VerticalAlignment="Top" Width="90"

```

```

Height="30" AutomationProperties.Name="Recovery_Button" Click="Recovery_Button_Click"
/>
    <Button x:Name="Avariya_Button" Content="Аварія"
HorizontalAlignment="Right" Margin="0,160,20,0" VerticalAlignment="Top" Width="60"
Height="30" AutomationProperties.Name="Avariya" Click="Avariya_Button_Click"/>
    <Label Content= "Режими роботи" HorizontalAlignment="Right"
Margin="0,130,40,0" VerticalAlignment="Top" AutomationProperties.Name="RegimLB"/>
    <Button x:Name="buttonToExcel" HorizontalAlignment="Right"
Margin="0,125,190,0" VerticalAlignment="Top" Width="60" Height="30"
AutomationProperties.Name="buttonToExcel">
        <Button.Background>
            <ImageBrush ImageSource="excel.png"/>
        </Button.Background>
    </Button>
    <Label Content= "Експорт to EXCEL" HorizontalAlignment="Right"
Margin="0,100,170,0" VerticalAlignment="Top" AutomationProperties.Name="RegimLB"/>
    <Button x:Name="EkranPB_click" HorizontalAlignment="Right" Height="280"
Margin="0,0,0,0" VerticalAlignment="Top" Width="450"
AutomationProperties.Name="EkranPB_click" IsHitTestVisible="True" ClipToBounds="True"
MouseMove="EkranPB_click_MouseMove" Visibility="Hidden" >
        <Button.Background>
            <ImageBrush ImageSource="ekran01.png"/>
        </Button.Background>
    </Button>
    <Image x:Name="EkranPB" HorizontalAlignment="Right" Height="280"
Margin="0,0,0,0" VerticalAlignment="Top" Width="450" Source="ekran007.png "
AutomationProperties.Name="EkranPB" Visibility="Hidden" />
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart1"
HorizontalAlignment="Left" Margin="6,289,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart2"
HorizontalAlignment="Left" Margin="398,285,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart3"
HorizontalAlignment="Left" Margin="184,289,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart4"
HorizontalAlignment="Left" Margin="826,285,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart5"
HorizontalAlignment="Left" Margin="661,501,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart6"
HorizontalAlignment="Left" Margin="430,501,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart7"
HorizontalAlignment="Left" Margin="621,306,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart8"
HorizontalAlignment="Left" Margin="220,501,0,0" VerticalAlignment="Top" Width="160"/>
    <OPCWPFDashboard:OPCWPFBarChart x:Name="chart9"
HorizontalAlignment="Left" Margin="20,501,0,0" VerticalAlignment="Top" Width="160"/>
</Grid>
</StackPanel>
</Window>

```

//-----

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PichWPF
{
    public static class DRand
    {
        public static double NextDouble(this Random random, double minValue, double
maxValue)
        {
            return random.NextDouble() * (maxValue - minValue) + minValue;
        }
    }
}

```

}

```
//-----

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
//using System.Drawing.Design;
using Microsoft.Office.Interop.Excel;
using Excel = Microsoft.Office.Interop.Excel;
using Application = Microsoft.Office.Interop.Excel.Application;
using System.Diagnostics;

using System.Windows.Forms;
using System.Windows.Forms.Integration;

namespace PichWPF
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    ///
    using MyCharts = System.Windows.Forms.DataVisualization.Charting.Chart;
    public partial class MainWindow : System.Windows.Window
    {
        List<ChartValues> _LstChartValues = null;
        List<MyCharts> _LstCharts = null;
        int img = 0;
        bool direc = false;
        bool CV = false;
        string _path = String.Empty;
        int ticksCout = 0;
        int ticksMoney = 0;
        double ampCount = 0;
        int i = 0;
        int N;
        float amper = 0;
        float volt = 0;
        double power = 0;
        double cost = 0;
        double cost1 = 0;
        double howmuch = 0;
        Random r = new Random();
        double v1 = 0;
        double v2 = 0;
        double v3 = 0;
        double T1 = 0;
        double T2 = 0;
        double T3 = 0;
        double Perepad = 0;
    }
}
```

```

double rnd = 0;
double rnd1 = 0;
double rnd2 = 0;
double rnd3 = 0;
double rnd4 = 0;
double rnd5 = 0;
double rnd6 = 0;
double volt2 = 0;
double power2 = 0;
double energy = 0;
double energy1 = 0;
Excel.Application xlApp = new Excel.Application();
System.Windows.Threading.DispatcherTimer t = new Sys-
tem.Windows.Threading.DispatcherTimer();
System.Windows.Threading.DispatcherTimer t1 = new Sys-
tem.Windows.Threading.DispatcherTimer();
System.Windows.Threading.DispatcherTimer t2 = new Sys-
tem.Windows.Threading.DispatcherTimer();
System.Windows.Threading.DispatcherTimer t3 = new Sys-
tem.Windows.Threading.DispatcherTimer();
System.Windows.Threading.DispatcherTimer t4 = new Sys-
tem.Windows.Threading.DispatcherTimer();
System.Windows.Threading.DispatcherTimer t5 = new Sys-
tem.Windows.Threading.DispatcherTimer();

```

```

public MainWindow()
{
    InitializeComponent();
    GreatPB.Visibility = Visibility.Hidden;

    t.Tick += new EventHandler(T_Tick);
    // t.Interval = new TimeSpan(0, 0, 1);
    t.Interval = new TimeSpan(5000000);

    t1.Tick += new EventHandler(T1_Tick);
    t1.Interval = new TimeSpan(1000000);

    t2.Tick += new EventHandler(T2_Tick);
    t2.Interval = new TimeSpan(1000000);

    t3.Tick += new EventHandler(T3_Tick);
    t3.Interval = new TimeSpan(5000000);

    t4.Tick += new EventHandler(T4_Tick);
    t4.Interval = new TimeSpan(1000000);

    t5.Tick += new EventHandler(T5_Tick);
    t5.Interval = new TimeSpan(2000000);
}

```

```

private void LoadChartsToList()
{
    _LstCharts = new List<MyCharts>();

    _LstCharts.Add(chart1);
    _LstCharts.Add(chart2);
    _LstCharts.Add(chart3);
    _LstCharts.Add(chart4);
    _LstCharts.Add(chart5);
    _LstCharts.Add(chart6);
    _LstCharts.Add(chart7);
    _LstCharts.Add(chart8);
    _LstCharts.Add(chart9);
}

```

```

    }

    private void StopAllTier()
    {
        t.Stop();
        t1.Stop();
        t2.Stop();
        t3.Stop();
        t4.Stop();
        t5.Stop();
    }

    private void ClearCharts()
    {
        v1 = 0;
        v2 = 0;
        v3 = 0;
        Perepad = 0;
        T1 = 0;
        T2 = 0;
        T3 = 0;
        ampCount = 0;
        power2 = 0;
        cost = 0;
        power = 0;
        cost1 = 0;
        N = 0;
        _LstChartValues.Clear();
        chart1.Series[0].Points.Clear();
        chart2.Series[0].Points.Clear();
        chart3.Series[0].Points.Clear();
        chart4.Series[0].Points.Clear();
        chart5.Series[0].Points.Clear();
        chart6.Series[0].Points.Clear();
        chart7.Series[0].Points.Clear();
        chart8.Series[0].Points.Clear();
        chart9.Series[0].Points.Clear();
    }

    private void ChartVisible(bool CV)
    {
        if (CV == true)
        {
            chart1.Visibility = Visibility.Visible;
            chart2.Visibility = Visibility.Visible;
            chart3.Visibility = Visibility.Visible;
            chart4.Visibility = Visibility.Visible;
            chart5.Visibility = Visibility.Visible;
            chart6.Visibility = Visibility.Visible;
            chart7.Visibility = Visibility.Visible;
            chart8.Visibility = Visibility.Visible;
            chart9.Visibility = Visibility.Visible;
        }
        else
        {
            chart1.Visibility = Visibility.Hidden;
            chart2.Visibility = Visibility.Hidden;
            chart3.Visibility = Visibility.Hidden;
            chart4.Visibility = Visibility.Hidden;
            chart5.Visibility = Visibility.Hidden;
            chart6.Visibility = Visibility.Hidden;
            chart7.Visibility = Visibility.Hidden;
            chart8.Visibility = Visibility.Hidden;
            chart9.Visibility = Visibility.Hidden;
        }
    }
}

```

```

private void DrawChart(int N)
{
    for (int i = 0; i < 9; i++)
    {
        if (i < 3)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0, N);
            _LstCharts[i].Series[0].Points.AddXY(N, GetV(i));
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i + 1).ToString(),
N.ToString(), GetV(i).ToString()));
        }
        else if (i >= 3 & i < 6)
        {
            _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0, N);
            _LstCharts[i].Series[0].Points.AddXY(N, GetT(i));
            _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
            _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
            _LstChartValues.Add(new ChartValues((i + 1).ToString(),
N.ToString(), GetT(i).ToString()));
        }
        else
        {
            if (i == 6)
            {
                _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);
                _LstCharts[i].Series[0].Points.AddXY(N, Perepad);
                _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
                _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
                _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), Perepad.ToString()));
            }
            if (i == 7)
            {
                _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);
                _LstCharts[i].Series[0].Points.AddXY(N, volt2);
                _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
                _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
                _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), volt2.ToString()));
            }
            if (i == 8)
            {
                _LstCharts[i].ChartAreas[0].AxisX.ScaleView.Zoom(0,
N);
                _LstCharts[i].Series[0].Points.AddXY(N, ampCount);
                _LstCharts[i].ChartAreas[0].AxisX.Minimum = N - 1;
                _LstCharts[i].ChartAreas[0].AxisX.Maximum = N;
                _LstChartValues.Add(new ChartValues((i +
1).ToString(), N.ToString(), ampCount.ToString()));
            }
        }
    }
}

private double GetT(int i)
{
    double retVal = 0;
    switch (i)
    {
        case 3:
            retVal = T1;
    }
}

```

```

        break;
    case 4:
        retVal = T2;
        break;
    case 5:
        retVal = T3;
        break;
    }
    return retVal;
}

private double GetV(int i)
{
    double retVal = 0;
    switch (i)
    {
        case 0:
            retVal = v1;
            break;
        case 1:
            retVal = v2;
            break;
        case 2:
            retVal = v3;
            break;
    }
    return retVal;
}

private void StartButton_Click(object sender, RoutedEventArgs e)
{
    t.Start();
    GreatPB.Visibility = Visibility.Hidden;
    StopButton.Visibility = Visibility.Visible;
    StartButton.Visibility = Visibility.Hidden;
    IndicatorLB.Content = "Індикатор стану печі. Наразі: Піч проходить
підготовку";
    IndicatorGrafCompleat.Content = "Індикатор стану печі. Наразі: Піч
проходить підготовку";
    CurrentTB.Visibility = Visibility.Visible;
    VoltageTB.Visibility = Visibility.Visible;
    PowerTB.Visibility = Visibility.Visible;
    MoneyPB.Source = new BitmapImage (new Uri (Directory.GetCurrentDirectory() + @"\Pich\1\dolar1.png"));
    progressBar1.Value = 0;
    SSEkrPB.Visibility = Visibility.Visible;
    SSEkrPB.Source = new BitmapImage (new
Uri (Directory.GetCurrentDirectory() + @"\Pich\1\start1.png"));
    Recovery_Button.Visibility = Visibility.Hidden;
}

private void T_Tick(object sender, EventArgs e)
{
    ticksCout++;
    if (img <= 7 & !direc)
    {
        amper = r.Next(2, 6);
        CurrentTB.Content = amper.ToString();
        volt = r.Next(219, 222);
        VoltageTB.Content = volt.ToString();
        power = (amper * volt);
        PowerTB.Content = (power).ToString();

        _path = Directory.GetCurrentDirectory() + @"\" + Pich + img + ".png";
        img++;
        //pictureBox1.Image = Image.FromFile(_path);
        pictureBox1.Source = new BitmapImage(new Uri (_path));
        if (img == 6)

```

```

        {
            direc = true;
        }
    }
    if (img > 0 & direc)
    {
        amper = r.Next(2, 6);
        CurrentTB.Content = amper.ToString();
        volt = r.Next(219, 222);
        VoltageTB.Content = volt.ToString();
        power = (amper * volt);
        PowerTB.Content = (power).ToString();

        _path = Directory.GetCurrentDirectory() + @"\" + Pich + img + ".png";
        img--;
        pictureBox1.Source = new BitmapImage(new Uri(_path));

        if (img == 0)
            direc = false;
    }

    if (ticksCout == 20)
    {
        t3.Start();
        StrON.Visibility = Visibility.Visible;
        IndicatorLB.Content = "Індикатор стану печі. Наразі: Печ
готова до процесу";
        IndicatorGrafCompleat.Content = "Індикатор стану печі. Наразі:
Піч готова до процесу";
        t.Stop();
        ticksCout = 0;
        pictureBox1.Source = new BitmapImage(new
Uri(Directory.GetCurrentDirectory() + @"\" + Pich + "1\8.png"));
        StartGrafit.Visibility = Visibility.Visible;
        power = (amper * volt2);
        PowerTB.Content = (power).ToString();
        SSEkrPB.Source = new BitmapImage (new
Uri(Directory.GetCurrentDirectory() + @"\" + Pich + "1\pause.png"));
        StrON.Source = new BitmapImage (new
Uri(Directory.GetCurrentDirectory() + @"\" + Pich + "1\arrow.png"));
        EkranPB.Visibility = Visibility.Hidden;
        EkranPB_click.Visibility = Visibility.Visible;
    }
}

private void T1_Tick(object sender, EventArgs e)
{
    if (img <= 4 & !direc)
    {
        _path = Directory.GetCurrentDirectory() +
@"\" + Pich + "2\" + img + ".png";
        img++;
        pictureBox1.Source = new BitmapImage(new Uri(_path));

        if (img == 3)
        {
            direc = true;
        }
    }
    if (img > 0 & direc)
    {
        _path = Directory.GetCurrentDirectory() +
@"\" + Pich + "2\" + img + ".png";
        img--;
    }
}

```



```

        pictureBox1.Source = new BitmapImage(new Uri(_path));

        if (img == 0)
            direc = false;
    }
}

private void T2_Tick(object sender, EventArgs e)
{
    //if (i == 0)
    //{
        ticksMoney++;
        MoneyPB.Location = new Sys-
tem.Drawing.Point(MoneyPB.Location.X + 10, MoneyPB.Location.Y);
        // if (MoneyPB.Location.X >= Width - MoneyPB.Width - 150) // и
здесь нужно подобрать правильно отступы
        // i++;

        // return;
    //}
    //if (i == 1)
    //{
        MoneyPB.Location = new Sys-
tem.Drawing.Point(MoneyPB.Location.X - 10, MoneyPB.Location.Y);
        // if (MoneyPB.Location.X <= 737) //здесь
        // i = 0;
        // return;
    //}
}

private void T3_Tick(object sender, EventArgs e)
{
    //if (i == 0)
    //{
        StrON.Location = new System.Drawing.Point(StrON.Location.X +
10, StrON.Location.Y);
        // if (StrON.Location.X >= Width - StrON.Width - 850) // и здесь
нужно подобрать правильно отступы
        // i++;
        // return;
    //}
    //if (i == 1)
    //{
        StrON.Location = new System.Drawing.Point(485,
StrON.Location.Y);
        // i = 0;
        // return;
    //}
}

private void T4_Tick(object sender, EventArgs e)
{
    N++;
    DrawChart(N);
    progressBar1.Value += 0.4;
    if (progressBar1.Value == 220)
    {
        progressBar1.Value = 0;
    }
    volt2 = r.Next(180, 230);
    VoltageCL.Content = (volt2).ToString();
    AmperCL.Content = Math.Round((ampCount += 0.2), 3).ToString();
    CurrentTB.Content = Math.Round((ampCount += 0.2), 3).ToString();
    VoltageTB.Content = Math.Round((volt2), 3).ToString();

    if (ampCount >= 88)
    {
        progressBar1.Value = 220;
        StopAllTier();
    }
}

```

```

IndicatorGrafCompleat.Content      =      "Вітаємо.      Графітування
завершене";

IndicatorLB.Content = "Вітаємо. Графітування завершене";
//MoneyPB.Location      =      new      System.Drawing.Point(376,
MoneyPB.Location.Y);
GreatPB.Visibility = Visibility.Visible;
GreatPB.Source      =      new      BitmapImage(new
Uri(Directory.GetCurrentDirectory() + @"\Pich\1\great.png"));
SSEkrPB.Source      =      new      BitmapImage      (new
Uri(Directory.GetCurrentDirectory() + @"\Pich\1\stop.png"));
    }
    var n = 220;
    var a = new double[n];
    var b = new double[n];
    var c = new double[n];
    var d = new double[n];
    var f = new double[n];
    var g = new double[n];
    var h = new double[n];
    for (var i = 0; i < n; i++)
    a[i] = r.NextDouble(-0.13, 1.24);
    b[i] = r.NextDouble(0.43, 1.34);
    c[i] = r.NextDouble(0.23, 1.14);
    d[i] = r.NextDouble(4.15, 5.15);
    f[i] = r.NextDouble(-0.95, 3.05);
    g[i] = r.NextDouble(3.01, 3.06);
    h[i] = r.NextDouble(3.09, 4.07);

    rnd = a[i];
    rnd1 = b[i];
    rnd2 = c[i];
    rnd3 = d[i];
    rnd4 = f[i];
    rnd5 = g[i];
    rnd6 = h[i];

    v1 += Math.Round(rnd, 3);
    v2 += Math.Round(rnd1, 3);
    v3 += Math.Round(rnd2, 3);
    Perepad += Math.Round(rnd3, 3);
    T1 += Math.Round(rnd4, 3);
    T2 += Math.Round(rnd5, 3);
    T3 += Math.Round(rnd6, 3);
    howmuch = Convert.ToDouble(howmuchTB.Text);

    power2 = Math.Round((ampCount * volt2), 3);
    energy = Math.Round((power2 / 3600000), 3);
    cost = energy * howmuch;
    MoneyLB.Content = (cost1 += cost).ToString();
    EnergyCL.Content = (energy1 += energy).ToString();
    PowerCL.Content = power2.ToString();
    PowerTB.Content = power2.ToString();
}

public void T5_Tick(object sender, EventArgs e)
{
    pictureBox1.Visibility = Visibility.Hidden;
    GreatPB.Visibility = Visibility.Visible;
    GreatPB.Source      =      new      BitmapImage      (new
Uri(Directory.GetCurrentDirectory() + @"\Pich\1\peopl1.png"));
    //GreatPB.Location = new System.Drawing.Point(GreatPB.Location.X -
10, GreatPB.Location.Y);
    //if (GreatPB.Location.X <= 250) // и здесь нужно подобрать пра-
вильно отступы
    //{
    //    GreatPB.Visibility = Visibility.Hidden;
    //    t5.Stop();

```

```

        //    pictureBox1.Visibility = Visibility.Visible;
        //    pictureBox1.Source = new BitmapImage (new
Uri (Directory.GetCurrentDirectory() + @"\Pich\1\Avariya1.png"));
        //    GreatPB.Location = new System.Drawing.Point (516,
GreatPB.Location.Y);
        //}
        ///MoneyPB.Location.X >= Width - MoneyPB.Width - 220
    }

    private void StartGrafit_Click(object sender, RoutedEventArgs e)
    {
        t4.Start();
        t2.Start();
        t1.Start();
        MoneyPB.Visibility = Visibility.Visible;
        StartGrafit.Visibility = Visibility.Hidden;
        IndicatorLB.Content = "Індикатор стану печі. Наразі: Піч графіта-
ції працює у робочому режимі";
        IndicatorGrafCompleat.Content = "Індикатор стану печі. Наразі: Піч
графітації працює у робочому режимі";
        GreatPB.Visibility = Visibility.Hidden;
        Cleaner(xlApp, false);
        SSEkrPB.Source = new BitmapImage (new
Uri (Directory.GetCurrentDirectory() + @"\Pich\1\start1.png"));
        ChartVisible(CV = true);
    }

    private void Stop_button2_Click(object sender, RoutedEventArgs e)
    {
        StopAllTier();
        ampCount = 0;
        pictureBox1.Source = new BitmapImage (new
Uri (Directory.GetCurrentDirectory() + @"\Pich\0.png"));
        StartButton.Visibility = Visibility.Visible;
        StopButton.Visibility = Visibility.Hidden;
        IndicatorLB.Content = "Індикатор стану печі. Наразі: Піч зупинена
:(";
        IndicatorGrafCompleat.Content = "Індикатор стану печі. Наразі: Піч
зупинена :(";
        StartGrafit.Visibility = Visibility.Hidden;
        MoneyPB.Visibility = Visibility.Hidden;
        StrON.Visibility = Visibility.Hidden;
        EkranPB_click.Visibility = Visibility.Visible;
        EkranPB.Visibility = Visibility.Visible;
        GreatPB.Visibility = Visibility.Hidden;
        StartGrafit.Visibility = Visibility.Visible;
        SSEkrPB.Source = new BitmapImage (new
Uri (Directory.GetCurrentDirectory() + @"\Pich\1\stop.png"));
        ClearCharts();
        ChartVisible(CV = false);
    }

    private void howmuchTB_KeyPress(object sender, KeyPressEventArgs e)
    {
        if ((e.KeyChar >= '0') && (e.KeyChar <= '9'))
        {
            return;
        }
        if (e.KeyChar == '.')
        {
            e.KeyChar = ',';
        }
        if (e.KeyChar == ',')
        {
            if (howmuchTB.Text.IndexOf(',') != -1)
            {

```

```

        e.Handled = true;
    }
    return;
}
if (Char.IsControl(e.KeyChar))
{
    if (e.KeyChar == (char)Keys.Enter)
        StartGrafit.Focus();
    return;
}
e.Handled = true;

//if (!Char.IsDigit(e.KeyChar) && e.KeyChar != 8)
//    e.Handled = true;
}

private void Cleaner(Excel.Application xlApp, bool KillProcess)
{
    try
    {
        //Excel object
        if (xlApp != null)
        {
            xlApp.Quit();
            xlApp = null;
        }
    }
    catch (Exception)
    {
    }

    //Excel Process
    if (KillProcess)
    {
        Process[] prc = Process.GetProcesses();
        for (byte i = 0; i < prc.Length; i++)
            if (prc[i].ProcessName.Equals("EXCEL"))
                prc[i].Kill();
    }

    //clear temp links and junk
    System.GC.Collect();
}

private void StopButton_Click(object sender, RoutedEventArgs e)
{
    StopAllTier();
    ampCount = 0;
    pictureBox1.Source = new BitmapImage(new
Uri(Directory.GetCurrentDirectory() + @"\Pich\0.png"));
    StartButton.Visibility = Visibility.Visible;
    StopButton.Visibility = Visibility.Hidden;
    IndicatorLB.Content = "Індикатор стану печі. Наразі: Піч зупинена
:(";
    IndicatorGrafCompleat.Content = "Індикатор стану печі. Наразі: Піч
зупинена :(";
    StartGrafit.Visibility = Visibility.Hidden;
    MoneyPB.Visibility = Visibility.Hidden;
    StrON.Visibility = Visibility.Hidden;
    EkranPB_click.Visibility = Visibility.Visible;
    EkranPB.Visibility = Visibility.Visible;
    GreatPB.Visibility = Visibility.Hidden;
    SSEkrPB.Source = new BitmapImage(new
Uri(Directory.GetCurrentDirectory() + @"\Pich\1\stop.png"));

```

```

        ClearCharts();
        ChartVisible(CV = false);
    }

    private void Recovery_Button_Click(object sender, RoutedEventArgs e)
    {
        Avariya_Button.Visibility = Visibility.Visible;
        Recovery_Button.Visibility = Visibility.Hidden;
    }

    private void Avariya_Button_Click(object sender, RoutedEventArgs e)
    {
        StopAllTier();
        ampCount = 0;
        StartButton.Visibility = Visibility.Hidden;
        IndicatorLB.Content = "Індикатор стану печі. Наразі: Піч зупинена. Аварія";
        IndicatorGrafCompleat.Content = "Індикатор стану печі. Наразі: Піч зупинена. Аварія";
        StartGrafit.Visibility = Visibility.Hidden;
        MoneyPB.Visibility = Visibility.Hidden;
        StrON.Visibility = Visibility.Hidden;
        GreatPB.Visibility = Visibility.Hidden;
        SSEkrPB.Source = new BitmapImage (new Uri (Directory.GetCurrentDirectory() + @"\Pich\1\stop.png"));
        t5.Start();
        Avariya_Button.Visibility = Visibility.Hidden;
        Recovery_Button.Visibility = Visibility.Visible;
    }
}

    private void EkranPB_click_MouseMove(object sender, System.Windows.Input.MouseEventArgs e)
    {
        EkranPB_click.Visibility = Visibility.Hidden;
        t3.Stop();
        StrON.Visibility = Visibility.Hidden;
        MoneyPB.Visibility = Visibility.Visible;
    }

    private void buttonToExcel_Click(object sender, EventArgs e)
    {
        string fileName = Directory.GetCurrentDirectory() +
        @"\Pich\1\ExcelFile.xlsx"; //@"C:\Pich\1\ExcelFile.xlsx"; //
        Excel.Workbook xlWb = xlApp.Workbooks.Open(fileName);
        //відкриваємо Excel файл
        Excel.Worksheet xlSht = xlWb.Sheets[1]; //перший лист в файлі
        int iLastRow = xlSht.Cells[xlSht.Rows.Count,
        "A"].End[Excel.XlDirection.xlUp].Row; //остання заповнена строка в стовбці A
        Excel.Range xRange = xlSht.Range["A1", "N300"];
        xRange.NumberFormat = "@";

        xlSht.Cells[1, "A"].Value = "№пп";
        xlSht.Cells[1, "B"].Value = "v1";
        xlSht.Cells[1, "C"].Value = "v2";
        xlSht.Cells[1, "D"].Value = "v3";
        xlSht.Cells[1, "E"].Value = "t1";
        xlSht.Cells[1, "F"].Value = "t2";
        xlSht.Cells[1, "G"].Value = "t3";
        xlSht.Cells[1, "H"].Value = "Перепад1";
        xlSht.Cells[1, "I"].Value = "Напруга";
        xlSht.Cells[1, "J"].Value = "Сила струму";
        xlSht.Cells[1, "K"].Value = "Перепад2";
        xlSht.Cells[1, "L"].Value = "Перепад3";
        xlSht.Cells[1, "M"].Value = "Час запису";

        bool f = false;
        if (!f)
        { iLastRow++; f = true; }
    }
}

```

```

        for (int i = 0; i < _LstChartValues.Count; i++)
        {
            System.Diagnostics.Debug.WriteLine($"i =
{i.ToString()} ::: {_LstChartValues[i].ChartId} -
{_LstChartValues[i].ValueY}");
        }
        for (int k = 0; k < _LstChartValues.Count; k++)
        {
            xlSht.Cells[iLastRow, "M"].Value = DateTime.Now; ;
            xlSht.Cells[iLastRow, "A"].Value = (iLastRow -

1).ToString();

            switch (_LstChartValues[k].ChartId)
            {
                case "1":
                    xlSht.Cells[iLastRow, "B"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "2":
                    xlSht.Cells[iLastRow, "C"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "3":
                    xlSht.Cells[iLastRow, "D"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "4":
                    xlSht.Cells[iLastRow, "E"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "5":
                    xlSht.Cells[iLastRow, "F"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "6":
                    xlSht.Cells[iLastRow, "G"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "7":
                    xlSht.Cells[iLastRow, "H"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "8":
                    xlSht.Cells[iLastRow, "I"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "9":
                    xlSht.Cells[iLastRow, "J"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "10":
                    xlSht.Cells[iLastRow, "K"].Value =
_LstChartValues[k].ValueY;
                    break;
                case "11":
                    xlSht.Cells[iLastRow, "L"].Value =
_LstChartValues[k].ValueY;
                    iLastRow++;
                    break;
            }
        }
        xlWb.Close(true); //закрити і зберегти книгу
        xlApp.Quit();
        MessageBox.Show("Файл успішно збережено!");
    }
}

```

```

private void StopAllTier()
{
    t.Stop();
    t1.Stop();
    t2.Stop();
    t3.Stop();
    t4.Stop();
    t5.Stop();
}

private void RegimTrackBar_Scroll(object sender, EventArgs e)
{
    if (RegimeTrackBar.Value == 1)
    {
        StopAllTier();
        ampCount = 0;
        StartButton.Visible = false;
        IndicatorLB.Text = "Індикатор стану печі. Наразі: Піч
зупинена. Аварія";
        IndicatorGrafCompleat.Text = "Індикатор стану печі. Наразі: Піч
зупинена. Аварія";
        StartGrafit.Visible = false;
        MoneyPB.Visible = false;
        StrON.Visible = false;
        GreatPB.Visible = false;
        SSEkrPB.Image = Image.FromFile(Directory.GetCurrentDirectory()
+ @"\Pich\1\stop.png");
        t5.Start();
    }
    else
    {
        IndicatorLB.Text = "Натисність кнопку STOP для налаштування";
        IndicatorGrafCompleat.Text = "Натисність кнопку STOP для
налаштування";
    }
}

private void StopButton_Click(object sender, EventArgs e)
{
    StopAllTier();
    ampCount = 0;
    pictureBox1.Image = Image.FromFile(Directory.GetCurrentDirectory()
+ @"\Pich\0.png");
    StartButton.Visible = true;
    StopButton.Visible = false;
    IndicatorLB.Text = "Індикатор стану печі. Наразі: Піч зупинена
:(";
    IndicatorGrafCompleat.Text = "Індикатор стану печі. Наразі: Піч
зупинена :(";
    StartGrafit.Visible = false;
    MoneyPB.Visible = false;
    StrON.Visible = false;
    EkranPB_click.Visible = true;
    EkranPB.Visible = true;
    GreatPB.Visible = false;
    SSEkrPB.Image = Image.FromFile(Directory.GetCurrentDirectory() +
@"\Pich\1\stop.png");
    RegimeTrackBar.Value = 2;
    ClearCharts();
    ChartVisible(CV = false);
}
}
}

```